



# Red Hat JBoss Data Virtualization 6.3 ユーザーガイド 1 : Teiid Designer

---

開発者向け

Red Hat Customer Content  
Services



開発者向け

Red Hat Customer Content Services

## 法律上の通知

Copyright © 2016 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

本書では、Red Hat JBoss Data Virtualization の Teiid Designer プラグインの使用について説明します。

## 目次

|   |           |
|---|-----------|
| <b>第1章 注意事項</b> .....   | <b>5</b>  |
| 1.1. データのバックアップ   | 5         |
| 1.2. 変数名: EAP_HOME  | 5         |
| 1.3. 変数名: MODE  | 5         |
| 1.4. Red Hat のドキュメントサイト   | 5         |
| <b>第2章 本書で使用される主な用語</b> .....                                   | <b>6</b>  |
| 2.1. Teiid Designer とは  | 6         |
| 2.2. メタデータとは  | 6         |
| 2.3. メタデータモデル   | 6         |
| 2.4. ビジネスおよびテクニカルメタデータ  | 7         |
| 2.5. テクニカルメタデータ   | 7         |
| 2.6. ビジネスメタデータ  | 7         |
| 2.7. デザインタイムおよびランタイムメタデータ                                       | 8         |
| 2.8. デザインタイムメタデータ   | 8         |
| 2.9. ランタイムメタデータ   | 8         |
| 2.10. モデルとは   | 8         |
| 2.11. ソースおよびビューメタデータ  | 10        |
| 2.12. Compact Node Type Definition                              | 10        |
| 2.13. Java コンテンツリポジトリ   | 10        |
| 2.14. ModeShape Tools   | 10        |
| 2.15. 仮想データベース  | 10        |
| <b>パート I. Teiid Designer</b> .....                              | <b>12</b> |
| <b>第3章 Teiid Designer の紹介</b> .....                             | <b>13</b> |
| 3.1. Teiid Designer を使う理由                                       | 13        |
| 3.2. ソースメタデータのモデリング   | 13        |
| 3.3. ビューメタデータのモデリング   | 13        |
| <b>第4章 : モデル</b> .....  | <b>15</b> |
| 4.1. プロセスの説明  | 15        |
| 4.2. モデルクラスおよびタイプ   | 15        |
| 4.3. VDB の内容と構造   | 15        |
| 4.4. モデルのバリデーション  | 16        |
| 4.5. モデルのテスト  | 16        |
| <b>第5章 モデルオブジェクトの拡張</b> .....                                   | <b>18</b> |
| 5.1. モデルオブジェクトの拡張   | 18        |
| 5.2. モデル拡張定義 (MED)  | 18        |
| 5.3. モデル拡張定義レジストリー (MED レジストリー)                                 | 19        |
| <b>第6章 サーバー管理</b> .....   | <b>20</b> |
| 6.1. JBoss Data Virtualization Server のターゲット                    | 20        |
| 6.2. サーバーの設定  | 22        |
| 6.3. JBDS をリモートの Red Hat JBoss Data Virtualization Server へ接続する | 25        |
| <b>第7章 Teiid Designer の例</b> .....                              | <b>27</b> |
| 7.1. Teiid Designer の例  | 27        |
| 7.2. Guides ビュー   | 27        |
| 7.3. JDBC ソースのモデリング   | 27        |
| 7.4. Cheat Sheets   | 40        |
| 7.5. SOAP Web サービスの消費   | 40        |

|  |            |
|--|------------|
| <b>第8章 New Model ウィザード</b> .....                   | <b>54</b>  |
| 8.1. New Model ウィザードの起動                            | 54         |
| 8.2. 新しいリレーショナルソースモデルの作成                           | 54         |
| 8.3. 新しいリレーショナルビューモデルの作成                           | 55         |
| 8.4. 新しい XML ドキュメントビューモデルの作成                       | 56         |
| 8.5. 新しい XML スキーマモデルの作成                            | 58         |
| 8.6. 新しい Web サービスビューモデルの作成                         | 59         |
| <b>第9章 インポーター</b> .....                            | <b>61</b>  |
| 9.1. インポーター  | 61         |
| 9.2. DDL のインポート                                    | 61         |
| 9.3. JDBC データベースからのインポート                           | 63         |
| 9.4. Teiid データソース接続からのインポート                        | 68         |
| 9.5. フラットファイルソースからのインポート                           | 69         |
| 9.6. XML データファイルソースからのインポート                        | 79         |
| 9.7. Salesforce からのインポート                           | 85         |
| 9.8. テキストファイルからのメタデータのインポート                        | 90         |
| 9.9. WSDL のリレーショナルモデルへのインポート                       | 99         |
| 9.10. WSDL の Web サービスへのインポート                       | 103        |
| 9.11. REST サービスからのデータのインポート                        | 120        |
| 9.12. LDAP (Lightweight Directory Access Protocol) | 121        |
| 9.13. LDAP サーバーからのインポート                            | 122        |
| 9.14. LDAP コネクタの更新機能                               | 123        |
| 9.15. XSD スキーマファイル                                 | 124        |
| 9.16. Red Hat JBoss Data Grid と複雑なデータタイプ           | 128        |
| <b>第10章 モデルオブジェクトの作成および編集</b> .....                | <b>129</b> |
| 10.1. 新しいモデルオブジェクトの作成                              | 129        |
| 10.2. Model Object Editors                         | 135        |
| 10.3. Transformation Editor                        | 137        |
| 10.4. Input Set Editor (XML)                       | 150        |
| 10.5. Choice Editor (XML)                          | 152        |
| 10.6. Recursion Editor (XML)                       | 154        |
| 10.7. Operation Editor                             | 157        |
| 10.8. モデルオブジェクト拡張の管理                               | 159        |
| <b>第11章 ユーザー定義関数</b> .....                         | <b>166</b> |
| 11.1. ユーザー定義関数                                     | 166        |
| 11.2. 関数のモデル化                                      | 166        |
| 11.3. トランスフォーメーションでの関数の使用                          | 167        |
| 11.4. VDB に関数を含める                                  | 167        |
| <b>第12章 メタデータ固有のモデリング</b> .....                    | <b>169</b> |
| 12.1. リレーショナルソースのモデリング                             | 169        |
| 12.2. リレーショナルビューのモデリング                             | 176        |
| 12.3. XML ドキュメントのモデリング                             | 192        |
| 12.4. Web サービスのモデリング                               | 194        |
| <b>第13章 モデルおよびプロジェクトの編集</b> .....                  | <b>207</b> |
| 13.1. モデルおよびプロジェクトの編集                              | 207        |
| 13.2. モデル名の変更                                      | 207        |
| 13.3. モデルの移動                                       | 208        |
| 13.4. モデルのコピーの保存                                   | 209        |
| 13.5. プロジェクトのクローン                                  | 211        |

|   |            |
|---|------------|
| <b>第14章 仮想データベースの管理</b> .....                                   | <b>214</b> |
| 14.1. 仮想データベースの作成   | 214        |
| 14.2. 仮想データベースの編集   | 214        |
| 14.3. 複数ソースのバインディングサポート   | 214        |
| 14.4. ユーザー定義関数のサポート   | 215        |
| 14.5. 仮想データベースの再使用  | 215        |
| 14.6. VDB ソースモデルの作成   | 216        |
| 14.7. セキュリティーおよびデータアクセス   | 217        |
| <b>第15章 モデルのテスト</b> .....                                       | <b>218</b> |
| 15.1. 接続プロファイルの管理   | 218        |
| 15.2. モデルのデータプレビュー  | 220        |
| 15.3. VDB を使ったテスト   | 225        |
| <b>第16章 検索</b> .....  | <b>232</b> |
| 16.1. 検索  | 232        |
| 16.2. モデルオブジェクトの検索  | 232        |
| 16.3. トランスフォーメーション SQL の検索                                      | 232        |
| 16.4. メタデータプロパティーによるモデルの検索                                      | 233        |
| <b>付録A サポートされる構成</b> .....                                      | <b>234</b> |
| A.1. サポートされるデータソースおよびトランスレーター                                   | 234        |
| A.2. JBoss Data Virtualization ランタイムでの Teiid Designer メタデータ使用要件 | 235        |
| <b>付録B ユーザー設定</b> .....   | <b>245</b> |
| B.1. ユーザー設定   | 245        |
| B.2. Teiid Designer の設定   | 245        |
| B.3. ダイアグラムの設定  | 246        |
| B.4. エディターの設定   | 247        |
| B.5. 検証の設定  | 250        |
| <b>付録C Teiid Designer UI 参照</b> .....                           | <b>253</b> |
| C.1. Teiid Designer パースペクティブ                                    | 253        |
| C.2. Teiid Designer パースペクティブ                                    | 253        |
| C.3. パースペクティブを開く  | 254        |
| C.4. 詳細情報   | 256        |
| <b>付録D Teiid Designer ビュー</b> .....                             | <b>257</b> |
| D.1. Teiid Designer ビュー   | 257        |
| D.2. Model Explorer ビュー   | 258        |
| D.3. 選択ベースのアクションメニュー  | 260        |
| D.4. Outline ビュー  | 261        |
| D.5. オンラインツリービュー  | 261        |
| D.6. アウトラインサムネイルビュー   | 262        |
| D.7. サーバービュー  | 263        |
| D.8. プロパティービュー  | 268        |
| D.9. 説明ビュー  | 270        |
| D.10. 問題ビュー   | 272        |
| D.11. ツールバーメニュー   | 273        |
| D.12. コンテキストメニュー  | 273        |
| D.13. 検索結果ビュー   | 274        |
| D.14. Datatype Hierarchy ビュー                                    | 276        |
| D.15. Teiid Model Classes ビュー                                   | 278        |
| D.16. Model Extension Registry ビュー                              | 278        |
| D.17. SQL Reserved Words ビュー                                    | 279        |

|  |            |
|--|------------|
| D.18. Model Extension Definition Registry ビュー (MED Registry ビュー) | 280        |
| D.19. Guides ビュー   | 281        |
| D.20. Status ビュー   | 283        |
| D.21. Cheat Sheets ビュー   | 284        |
| <b>付録E エディター</b> .....   | <b>285</b> |
| E.1. エディター   | 285        |
| E.2. Model Editor  | 287        |
| E.3. Table Editor  | 292        |
| E.4. Simple Datatypes Editor                                     | 297        |
| E.5. Semantic Editor   | 298        |
| E.6. Source Editor   | 298        |
| E.7. Model Object Editors  | 299        |
| E.8. VDB Editor  | 299        |
| E.9. Model Extension Definition Editor                           | 303        |
| <b>付録F Teiid Designer メインメニュー</b> .....                          | <b>307</b> |
| F.1. Teiid Designer メインメニュー                                      | 307        |
| F.2. File メニュー   | 307        |
| F.3. Edit メニュー   | 310        |
| F.4. Refactor メニュー   | 311        |
| F.5. Navigate メニュー   | 312        |
| F.6. Search メニュー   | 312        |
| F.7. Project メニュー  | 313        |
| F.8. Metadata メニュー   | 315        |
| F.9. Run メニュー  | 315        |
| F.10. Window メニュー  | 315        |
| F.11. Help メニュー  | 317        |
| <b>付録G 改訂履歴</b> .....  | <b>319</b> |



## 第1章 注意事項

### 1.1. データのバックアップ



#### 警告

Red Hat は、システム設定をバックアップしてから本書に掲載されている設定タスクを実行することを推奨します。

### 1.2. 変数名: EAP\_HOME

**EAP\_HOME** は、JBoss Data Virtualization がデプロイされた Red Hat JBoss Enterprise Application Platform インストールのルートディレクトリを示しています。

### 1.3. 変数名: MODE

**MODE** は、JBoss Data Virtualization が実行されているモードがスタンドアロンまたはドメインモードであるかによって **standalone** または **domain** のいずれかになります。本書に記載されているファイルパスに **MODE** がある場合は、これらのどちらかに置き換えてください (この変数は、ディレクトリ構造のどこに製品がインストールされているかに応じて、ご自身で設定する必要があります)。

### 1.4. Red Hat のドキュメントサイト

Red Hat の公式ドキュメントサイトは <https://access.redhat.com/site/documentation/> になります。本書を含む最新バージョンのドキュメントをご覧になれます。

## 第2章 本書で使用される主な用語

### 2.1. Teiid Designer とは

**Teiid Designer** は、ビジネスデータを公開するリレーショナル、XML、および Web サービスビューを生成するために、複数のデータソースをモデリング、分析、統合、およびテストする Eclipse ベースのグラフィカルモデリングツールです。

### 2.2. メタデータとは

メタデータとは、データに関するデータのことです。Teiid Designer では、特定の情報構造を構成する個々のデータフィールドに関係なく、メタオブジェクトと呼ばれるメタデータの一部にその構造に関する情報が含まれます。

基本的なデータベースの例として、アドレス帳について考えてみましょう。アドレス帳の中には、ZIP コード (郵便番号) のフィールドまたは列があるはずです。アドレス帳がアメリカ国内の住所に対応している場合、ZIP コードのフィールドまたは列に関して以下のことが推定されます。

- ※ 名前付きの ZIPCode
- ※ 数値
- ※ 文字列
- ※ 9 文字
- ※ StreetAddress テーブルにある
- ※ 2 つの部分で構成されます。最初の 5 桁の数字は 5 つの ZIP コード番号を表し、最後の 4 桁は ZIP プラス 4 の番号を表します (ZIP プラス 4 がない場合は 0000)。
- ※ 整数の数字のみにフォーマットされます。631410.00 や 6314q0000 などにフォーマットされるとエラーが発生します。

この定義は、アドレス帳データベースの ZIP コードデータに関するメタデータを表しています。これはデータベース自体から情報を抽象化します。エンタープライズ情報システムの内容を記述したり、あるエンタープライズ情報ソースの列がどのように別の列と関連し、これらの 2 つの列を他の目的でどのように一緒に使用できるかを判断したりするのに役立ちます。

このメタデータでは以下を考慮できます。

- ※ どのような情報がメタデータに含まれるか ([「ビジネスおよびテクニカルメタデータ」](#)を参照)
- ※ メタデータはどのデータを表すか ([「ソースおよびビューメタデータ」](#)を参照)
- ※ 自分の組織でどのようにこのメタデータを使用および管理するか ([「デザインタイムおよびランタイムメタデータ」](#)を参照)

### 2.3. メタデータモデル

メタデータモデルは、完全なデータ構造を示すメタデータ情報の集まりを表します。

前の例では、フィールド ZIPCode はアドレス帳データベースのメタデータオブジェクトであると説明しました。このメタデータオブジェクトは、メタデータ情報の 1 つの違いを表します。親テーブル StreetAddress についても触れました。これらのメタデータオブジェクトと、データベース内の他のテーブルや列を記述する他のメタデータオブジェクトをすべて組み合わせ、オブジェクトをすべてホストするエン

タープライズ情報システムのソースメタデータモデルが形成されます。

メタデータモデルのコレクション内にソースモデルを作成できます。これらは物理データストレージの場所を表します。また、データのビジネスビューをモデリングするビューモデルを作成することもできます。ソースとビューメタデータの違いについては、「ソースおよびビューメタデータ」の項を参照してください。

## 2.4. ビジネスおよびテクニカルメタデータ

メタデータは、データの一部に関する異なるタイプの情報を含むことができます。

- ※ テクニカルメタデータは、データの場所やネイティブ環境でのデータ構造など、データのアクセスに必要な情報を記述します。
- ※ ビジネスメタデータは、メタデータオブジェクトに関連するキーワードやメタデータオブジェクトに関する注記など、データに関する他の情報を記述します。



### 注記

テクニカルおよびビジネスメタデータという言葉は、メタデータの内容であるメタデータに含まれる情報のタイプを示します。メタデータが何を表すかを示す、物理およびビューメタデータという言葉と混同しないようにしてください。

## 2.5. テクニカルメタデータ

テクニカルメタデータは、元のネイティブデータストレージにあるデータへアクセスする方法を記述する情報を表します。技術メタデータには、データタイプ、エンタープライズ情報システムのデータの名前、およびネイティブのエンタープライズ情報システムがメタオブジェクトを識別する方法を記述するその他の情報が含まれます。

アドレス帳データベースの例で、ZIP コード列に関して認識している技術メタデータは次のとおりです。

- ※ 名前付きの ZIPCode
- ※ 9 文字
- ※ 文字列
- ※ StreetAddress テーブルにある
- ※ SQL クエリ言語を使用

これらの情報は、エンタープライズ情報システムのデータへアクセスし、処理するために必要なデータおよび情報を記述します。

## 2.6. ビジネスメタデータ

ビジネスメタデータは、エンタープライズ情報システムの物理ストレージまたはデータアクセス要件に関連しているとは限らないデータの一部に関する追加情報を表します。また、記述、ビジネスルール、およびデータの一部に関するその他の追加情報も表します。

アドレス帳データベースの例では、ZIP コードに関するビジネスメタデータを表すのは次のとおりです。

- ※ 最初の 5 文字は 5 つの ZIP コード番号を表し、最後の 4 桁は ZIP プラス 4 の番号を表します (ZIP プラ

ス 4 がない場合は 0000)。

- ※ データベースのこのフィールドを入力するために使用されるアプリケーションは、データ形式の整合性を厳格に強制します。

上記の最初の項目は技術的であるように見えますが、データの物理ストレージとは直接関連していません。列の内容に適用されるビジネスルールを表し、内容自体は表しません。

上記の 2 番目の項目は、列が事前設定される方法に関するビジネス情報を表します。この情報は、列の定義と関連付けると便利ですが、データの物理ストレージを反映しません。

## 2.7. デザインタイムおよびランタイムメタデータ

**Teiid Designer** ソフトウェアはデザインタイムメタデータとランタイムメタデータを区別します。Teiid Designer Server を使用する場合、この区別が重要になります。デザインタイムデータにはメタデータを理解し、効率的に整理するための詳細や表現が含まれています。詳細の多くは、作成する仮想データベースを実行する基盤のシステムには必要ありません。システムのパフォーマンスを最大限に引き出すため、仮想データベースの実行に必ず必要であるとは限らない情報はランタイムメタデータから削除されます。

## 2.8. デザインタイムメタデータ

デザインタイムメタデータとは、ローカルディレクトリー内で作成またはインポートされたデータのことです。このメタデータは、ソースおよびビューメタデータを追加し、**Teiid Designer** でモデリングできます。

## 2.9. ランタイムメタデータ

ソースの物理構造を記述する必要なテクニカルメタデータなど、エンタープライズ情報システムを適切にモデリングしたら、データアクセスのメタデータを使用できます。

Teiid Designer Server で使用するメタデータを準備するには、クライアントアプリケーションからクエリーを解決するときに、Teiid Designer Server で使用するメタデータモデルのスナップショットを作成します。このランタイムメタデータは、作成またはインポートしたデザインタイムメタデータの静的バージョンを表します。このスナップショットは、仮想データベース定義 (VDB) になります。

このランタイムメタデータの作成時に **Teiid Designer** では以下が実行されます。

- ※ メタデータモデルの一貫したセットからランタイムメタデータが派生されます。
- ※ 基盤のエンタープライズ情報システムへのアクセスを記述するテクニカルメタデータを重視し、デザインタイムメタデータのサブセットを作成します。
- ※ データアクセスのパフォーマンスに対し、ランタイムメタデータを最適化します。

デザインタイムメタデータを継続して使用できますが、ランタイムメタデータモデルの作成後は静的な状態を維持します。

## 2.10. モデルとは

モデルは、情報コンストラクトのセットを表します。列で構成され、データの記録が含まれるテーブルを定義するリレーショナルモデルは一般的なモデルの 1 つです。XML モデルも一般的なモデルで、階層データのセットを定義します。

**Teiid Designer** では、モデルはエンティティーの定義とエンティティー間の関係の定義に使用されます。これは、情報セットの統合を完全に定義し、単一の API とアクセスプロトコルを使用して同じ方法でアクセスされるようにするために必要です。これらのモデルに使用されるファイル拡張子は **.xmi** (例: **NorthwindOracle.xmi**) で、OMG によって定義される XMI 構文に準拠します。

以下はモデルファイルの部分的なコンテンツの例になります。

```
<?xml version="1.0" encoding="ASCII"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.omg.org/XMI http://www.omg.org/XMI.xmi">
  <mmcore:ModelAnnotation xmi:uuid="mmuuuid:b0355f00-413b-1079-9d18-8acf4a7127d5" description="Northwind_Oracle was created by Teiid Designer" modelLocation="http://www.w3.org/2001/XMLSchema#long" name="XMLESchema" modelLocation="http://www.w3.org/2001/XMLSchema#long">
    <modelImports xmi:uuid="mmuuuid:2d815780-4140-1079-9d18-8acf4a7127d5" name="XMLESchema" modelLocation="http://www.w3.org/2001/XMLSchema#long" nameInSource="XMLESchema" modelLocation="http://www.w3.org/2001/XMLSchema#long">
      <modelImports xmi:uuid="mmuuuid:2e663940-4140-1079-9d18-8acf4a7127d5" name="SimpleDatatypes-instance" modelLocation="http://www.w3.org/2001/XMLSchema#long" nameInSource="SimpleDatatypes-instance" modelLocation="http://www.w3.org/2001/XMLSchema#long">
        </modelImports>
      </modelImports>
    </mmcore:ModelAnnotation>
    <relational:BaseTable xmi:uuid="mmuuuid:b20e64c0-413b-1079-9d18-8acf4a7127d5" name="CATEGORIES" nameInSource="CATEGORIES" nativeType="http://www.w3.org/2001/XMLSchema#string">
      <columns xmi:uuid="mmuuuid:bb5ac3c0-413b-1079-9d18-8acf4a7127d5" name="CATEGORYID" nameInSource="CATEGORYID" nativeType="http://www.w3.org/2001/XMLSchema#long">
        </columns>
      <columns xmi:uuid="mmuuuid:bc4ee7c0-413b-1079-9d18-8acf4a7127d5" name="CATEGORYNAME" nameInSource="CATEGORYNAME" nativeType="http://www.w3.org/2001/XMLSchema#string">
        </columns>
      <columns xmi:uuid="mmuuuid:bc4ee7c1-413b-1079-9d18-8acf4a7127d5" name="DESCRIPTION" nameInSource="DESCRIPTION" nativeType="http://www.w3.org/2001/XMLSchema#string">
        </columns>
      <columns xmi:uuid="mmuuuid:bc4ee7c2-413b-1079-9d18-8acf4a7127d5" name="PICTURE" nameInSource="PICTURE" nativeType="http://www.metamatrix.com/metamodels/SimpleDatatypes-instance#blob">
        </columns>
      <primaryKey xmi:uuid="mmuuuid:d481f940-413b-1079-9d18-8acf4a7127d5" name="PK_CATEGORIES" nameInSource="PK_CATEGORIES">
        </primaryKey>
      </relational:BaseTable>
  </xmi:XMI>
```

図2.1 サンプルモデルファイル

### 注記

モデルファイルは手動で編集しないでください。手動で編集することは可能ですが、ファイルが破損し、**Teiid Designer** システム内で使用できなくなる可能性があります。

**Teiid Designer** の基本のモデルは、データソースに含まれる情報の構造およびデータ特性を定義します。これらはソースモデルと呼ばれます。**Teiid Designer** はソースモデルの情報を使用して、複数のソースの情報に対してフェデレーションを行うため、ユーザーの観点からは単一のソースであるかのように見えます。

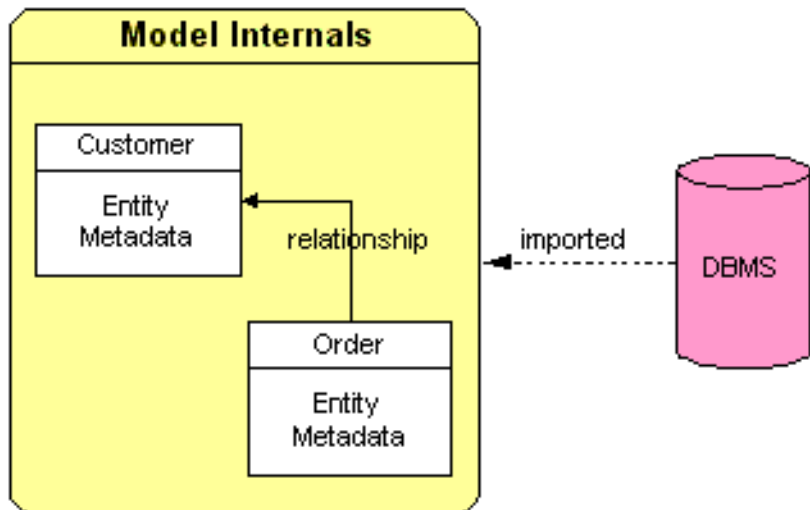


図2.2 モデルの内部

ソースモデルに加え、**Teiid Designer** はさまざまなビューモデルを定義する機能を提供します。これらのモデルは、物理(またはソース)レイヤーの上位にある抽象化のレイヤーを定義するために使用されるため、情報をエンドユーザーに提示でき、物理的に保存された状態のままではなく、ビジネス用語でアプリケーションを消費します。これらのビジネスビューはさまざまな形式を取ることができます。

※ リレーショナルテーブルおよびビュー

- ※ XML
- ※ Web サービス
- ※ Function パネル

3 つ目のモデルタイプは論理タイプで、論理または構造の観点からモデルを定義する機能を提供します。

## 2.11. ソースおよびビューメタデータ

ビジネスメタデータとテクニカルメタデータの違いの他に、ソースメタデータとビューメタデータの違いを知っておく必要があります。

ソースメタデータとビューメタデータとは、内容自体ではなくメタデータが表すものを指します。

ソースメタデータはエンタープライズ情報システムのメタデータを直接表し、データがどこでどのように維持されるかを正確にキャプチャーします。ソースメタデータはテクニカルメタデータと似ていると感じられるかもしれませんが、ソースメタデータにはテクニカルメタデータとビジネスメタデータの両方を含めることができます。ソースメタデータをモデリングする場合、エンタープライズ情報システムに含まれるデータをモデリングすることになります。

これに対し、ビューメタデータはソースメタデータを異なるアプリケーションの用語やドメインに変換する、調整されたビューを表します。ビューメタデータにもテクニカルメタデータとビジネスメタデータの両方を含めることができます。ビューメタデータをモデリングする場合、アプリケーション (および企業) が最終的に使用するようにデータをモデリングすることになります。

## 2.12. Compact Node Type Definition

Compact Node Type Definition (CND) はコンパクトな標準形式でノードを定義するファイル形式です。通常の CND ファイルには、名前空間の宣言、ノードタイプの定義、およびコメントが含まれます。CND は、ノードタイプの定義および名前空間の宣言に、コンパクトで標準的な構文を提供します。

## 2.13. Java コンテンツリポジトリ

Java コンテンツリポジトリ (JCR) は Java API のコンテンツリポジトリです。JCR は、情報に関する以下のサービスを提供します。

- ※ 情報の読み書きアクセス
- ※ 階層的にファイルを構造化する機能
- ※ 構造化されたコンテンツや構造化されていないコンテンツを使用して作業する機能
- ※ 検索およびクエリーを行う機能
- ※ 情報のバージョン管理
- ※ コンテンツへのアクセスを制御する機能

## 2.14. ModeShape Tools

ModeShape Tools は、ModeShape および Java コンテンツリポジトリ (JCR) を使った作業に使用できる Eclipse プラグインのセットです。

## 2.15. 仮想データベース

### 4.13. 仮想データベース

**Teiid Designer** が管理を目的する重要なアーティファクトは仮想データベース (VDB) です。VDB は JBoss Data Virtualization を介し、VDB がどのように設定されているかを基にして接続、クエリーおよびアップデートできる標準の JDBC データベーススキーマのように動作します。VDB はデプロイされるとデータベースになるため、他のビューモデルトランスフォーメーションへのソースとして使用できます。これにより、ビジネスの要件に応じて再使用可能な VDB や共通の VDB を複数のレイヤーに作成およびデプロイできます。

## パート I. Teiid Designer



## 第3章 Teiid Designer の紹介

### 3.1. Teiid Designer を使う理由

Teiid Designer を使用すると、データソースを対象の形式へマップできます。また、以下を行うことが可能になります。

- ※ セマンティックな差異を解決します。
- ※ 物理または論理レベルで仮想データ構造を作成します。
- ※ 宣言型インターフェースを使用して、ソースと対象の形式の間でデータを統合、集約、および変換します。これは、アプリケーションと互換性があり、アプリケーションによる消費に対して最適化されます。

これにより、基盤の物理データ構造から、アプリケーションで使用する情報の構造を抽象化できます。Teiid Designer を使用すると、データサービスが迅速に定義され、結果のアーティファクトの維持や再使用が簡単です。さらに、後で参照するために、貴重な作業や関連するメタデータが保存されます。

また、Teiid Designer を使用して複数のソースを統合し、以下の一般的なデータアクセス標準を使用してアクセスすることができます。

- ※ Web サービス/SOAP/XML
- ※ JDBC/SQL
- ※ ODBC/SQL

### 3.2. ソースメタデータのモデリング

エンタープライズ情報システム内でソースメタデータをモデリングする場合、以下を含む詳細な情報の一部がキャプチャーされます。

- ※ データタイプのアイデンティティ
- ※ ストレージの形式
- ※ 制約
- ※ ソース固有の場所および名前

ソースメタデータは、この詳細なテクニカルメタデータをキャプチャーし、データのマップ、データの場所、およびアクセス方法を提供します。

ソースメタデータのこのコレクションは、企業内の情報ソースの直接マッピングを構成します。情報の統合に Teiid Designer Server を使用する場合、このテクニカルメタデータはクエリーの解決に不可欠です。

たとえば、これまでの例を用いると、ZIPCode 列とその親テーブルである StreetAddress はアドレス帳データベース内のフィールドに直接マップします。

この例をさらに広げ、マーケティングリサーチ会社によって提供されるコマ区切りのテキストファイルが 2 番目の情報ソースとしてあるとします。このテキストファイルは、住所や ZIP コードを基にした追加の人口統計情報を提供できます。このテキストファイルは別のエンタープライズ情報システム (EIS) を表し、ソースモデルのメタオブジェクトは個々のコマ区切りの値を記述します。

### 3.3. ビューメタデータのモデリング

ビューメタデータを作成するときは、物理データストレージの性質を記述せずに、企業の日常業務でどのように情報を使用するかを記述します。

ビューメタデータのクラスと属性は他のメタデータから派生されます。メタデータの最終的なソースを記述するソースメタデータまたは他のビューメタデータからビューメタデータを派生できます。しかし、ビューメタデータをモデリングする場合、既存のエンタープライズ情報システム上に、事業またはアプリケーションに合わせて調整できる特別なビューを作成します。このビューメタデータには多くの利点があります。

- ※ アプリケーションに関する情報のみを公開できます。アプリケーションはこのビューメタデータを使用し、最終的な物理データストレージに対してクエリーを解決します。
- ※ ビューメタデータをアプリケーションが使用する既存のビューメタデータに追加すると、データの異なるビューが必要な既存アプリケーションにコンテンツを追加できます。新しいモデルを作成したり既存のアプリケーションを編集したりする必要がないため、時間と労力を節約できます。
- ※ アプリケーションが特定の物理エンタープライズ情報システムを参照する必要がないため、柔軟性や代替の可能性が向上されます。情報のソースを変更するため、アプリケーションを変更する必要がありません。
- ※ ビューメタデータモデルは、企業がデータを使用するさまざまな方法や、その情報を参照するさまざまな用語を記述します。これは一元的な場所で行われます。

エンタープライズ情報ソースの例では、アドレス帳データベースとベンダーが提供するコンマ区切りテキストファイルは、異なる 2 つのネイティブストレージ形式にあるため、2 つのソースメタデータモデルがあります。しかし、これらは大量のメールを送信するためのアドレスプールという 1 つのビジネスニーズのみを表すことができます。

ビューメタデータモデルを作成すると、単一のビューテーブルである AddressPool に 2 つのエンタープライズ情報システムからの情報が含まれることを明確に表すことができます。ビューメタデータモデルは、どこから情報を収集するかを表すだけでなく、ソースモデルから情報を選択するために実行する SQL 演算も表します。

このビューメタデータは、組織がどのように情報を使用するかを反映および記述できませんが、企業で Teiid Designer Server を使用すれば、アプリケーションはビューメタデータを使用してクエリーを解決できます。

このビューメタデータを作成するにはビューを作成します。さらに、ソース (または他のビュー) メタデータモデルから情報の選択を可能にする特別なクエリーである、そのビューのトランスフォーメーションを定義します。

## 第4章 : モデル

### 4.1. プロセスの説明

Teiid Designer を使用してモデルを構築するプロセスをできる限り簡単にするため、ガイドビュー ([「Guides ビュー」](#) を参照) が導入されました。これは、特定のユースケースに対してモデルを作成するために必要なアクションをまとめる、アクションのセットを提供します。アクションセットは以下のシナリオで利用可能です。

- ※ SOAP Web サービスの消費
- ※ フラットファイルソース (テキストファイル) からのモデリング
- ※ JDBC データソースからのモデリング
- ※ ローカル XML ファイルソースからのモデリング
- ※ リモート XML ファイルソースからのモデリング
- ※ Red Hat JBoss Data Virtualization データソース (サーバー上にデプロイ) からのモデリング
- ※ Red Hat JBoss Data Virtualization サーバーへの接続

### 4.2. モデルクラスおよびタイプ

Teiid Designer を使用すると、モデルのさまざまなクラスをモデリングできます。各クラスは、モデルの概念が異なる分類を表します。

- ※ リレーショナル: テーブル列とレコード形式で表すことができるモデルデータです。リレーショナルモデルは、リレーショナルデータベース、スプレッドシート、テキストファイル、または簡単な Web サービスにある構造を表すことができます。
- ※ XML: XML ドキュメントの基本的な構造を表すモデルです。XML スキーマによるサポートが可能です。XML モデルは再帰的な階層などのネストされた構造を表します。
- ※ XML スキーマ: XML ドキュメントの構造および内容を形式的に定義するための W3C 標準であり、XML ドキュメントで許可される値を定義するデータタイプです。
- ※ Web サービス: Web サービスインターフェース、演算、および演算入出力パラメーター (XML スキーマの形式) を定義します。
- ※ 関数: 関数メタモデルは、カスタムのトランスフォーメーション SQL ステートメントで使用する、バイナリソース jar などのユーザー定義の関数を提供する機能をサポートします。

### 4.3. VDB の内容と構造

Teiid Designer では、VDBファイル名に **.vdb** ファイル拡張子を使用します。VDB は構造上、3つのフォルダーが含まれる ZIP アーカイブファイルです。

- ※ META-INF: **vdb.xml** 定義ファイルが含まれます。
- ※ runtime-inf - VDB に含まれる各モデルのバイナリー INDEX ファイルが含まれます。
- ※ project folder: VDB エディターに追加するモデルが含まれます (**\*.xmi** および **\*.xsd** ファイル)。

デプロイすると、モデル定義に必要なランタイムメタデータを作成するため、メタデータは JBoss Data Virtualization によって消費されます。

vdb.xml ファイルには以下が含まれます。

- ※ VDB 名、バージョン、プロパティー
- ※ 含まれるモデル情報 (名前、トランスレーター名、接続情報)
- ※ トランスレーターの情報
- ※ 参照されたモデルのデータロール定義
- ※ VDB リファレンスのインポート

Teiid Designer は、一貫性のある有効な **vdb.xml** ファイルを維持する専用の VDB エディターを提供して VDB の管理を容易にし、ワークスペースモデルを VDB の関連するモデルと同期化できるようにします ([「VDB Editor」](#) を参照してください)。

## 4.4. モデルのバリデーション

モデルをデータのアクセスに使用するには、モデルが有効な状態である必要があります。単一モデルのバリデーションでは、欠如しているものや存在しないエンティティへの参照がなく、そのモデル自体が一貫し、完全な状態である必要があります。複数モデルのバリデーションでは、モデル間の依存関係がすべて存在し、解決可能であることをチェックします。

データアクセスの目的でモデルが VDB にデプロイされたときに必ずバリデーションを行う必要があります。Teiid Designer では、モデルが保存されたときに自動的にバリデーションが行われます。



### 注記

**Project > Build Automatically** メニューオプションを選択する必要があります。モデルを編集するときに保存されていない変更があると、エディタータブに \* が表示されます。

## 4.5. モデルのテスト

データの設計や作業を行うとき、データが見れると作業が大変楽になる場合がよくあります。Teiid Designer のプレビューデータ機能はこれを可能にし、物理テーブルや仮想ビューに関わらず、オブジェクトによって記述された情報のプレビューを即座に表示することができます。テーブル、ビュー、手順、または XML ドキュメントを選択するだけで、実際のデータでビューをテストできます。このプレビュー機能は、VDB がサーバーにデプロイされたときに、Teiid Designer でのデータアクセスの動作が確実に一致するようにします。

情報のプレビューは、データを試す簡単で迅速な方法です。アプリケーションが使用するようなより複雑なクエリーを実行するには、Teiid Designer で VDB を実行し、クエリーまたは SQL ステートメントを入力します。

モデルを作成した後、プレビューデータアクションを使用してテストできます。希望のテーブルオブジェクトを選択し、アクションを実行すると、簡単なクエリーの結果がデータツール SQL 結果ビューに表示されます。このアクションは、さまざまなビューツールバーおよびコンテキストメニューで Teiid Designer 全体にわたってアクセス可能です。

プレビュー可能なオブジェクトには以下が含まれます。

- ※ アクセスポターンが関係するテーブルを含む、リレーショナルテーブルまたはビュー。
- ※ リレーショナルプロシージャ
- ※ Web サービス演算
- ※ XML ドキュメントステージングテーブル



### 注記

リレーショナルアクセスポターン、Web サービス演算、または入力パラメーターを持つリレーショナルプロシージャのプレビューを表示しようとする、ダイアログによって必要なパラメーターの値が要求されます。

## 第5章 モデルオブジェクトの拡張

### 5.1. モデルオブジェクトの拡張

**Teiid Designer** は JBoss Data Virtualization とともに拡張可能なフレームワークを提供し、メタモデルに定義されたこと以外にモデルオブジェクトのカスタムプロパティを定義します。これらのカスタムプロパティ値は VDB に追加され、ランタイムメタデータに含まれます。この追加のメタデータは、ソースクエリーの操作と返された結果セットデータの調整に対してカスタムトランスレーターで使用できます。

**Teiid Designer** は、EMF ベースのモデル拡張メタモデルに代わる新しいモデル拡張定義 (MED) フレームワークを導入します。

この MED フレームワークによる改善点は次のとおりです。

- ※ 別の EMF メタモデルは必要ありません。
- ※ 拡張可能なメタデータモデルおよびメタモデルオブジェクトの削減 (リレーショナル、Web サービス、XML ドキュメント、ユーザー定義の関数) を含め、EMF の用語を基本オブジェクトタイプに置き換える方法がより簡単です。
- ※ 複数の MED によるメタモデルの拡張が可能です。
- ※ MED はモデルに保存されるため、VDB には追加の依存関係が必要ありません。

[「モデルオブジェクト拡張の管理」](#) および [「Model Extension Definition Editor」](#) も参照してください。

### 5.2. モデル拡張定義 (MED)

MED の目的は、拡張プロパティのセットを 1 つ以上定義することです。拡張プロパティの各セットは、1 つのモデルオブジェクトタイプ (またはメタクラス) に関連します。各 MED は以下によって構成されます。

- ※ 名前空間の接頭辞: 一意の識別子です。通常は少数の文字のみで、名前空間 URI の省略形として使用できます。
- ※ 名前空間 URI: 一意の URI です。
- ※ 拡張メタデータモデル URI (モデルクラス): 拡張されたメタデータモデル URI です。各メタデータモデル URI にはモデルクラスもあり、通常はこれが Designer で表示されます。拡張に対してサポートされるモデルクラスは、リレーショナル、Web サービス、XML ドキュメント、および関数です。
- ※ バージョン (現在使用されていません)。
- ※ 記述: オプションの記述および目的
- ※ 拡張モデルオブジェクトタイプ (メタクラス): 拡張プロパティが定義されているモデルオブジェクトタイプまたはメタクラスのセット。
- ※ プロパティ: モデルオブジェクトタイプによってグループ化される拡張プロパティ定義。

MED ファイルは拡張子が **mxid** の XML ファイルです。MED スキーマファイル (添付の **modelExtension.xsd** を参照) は MED ファイルの検証に使用されます。以下に MED ファイルの例を示します。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<modelExtension xmlns:p="http://org.teiid.modelExtension/2011"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
metamodelUri="http://www.metamatrix.com/metamodels/Relational"
namespacePrefix="mymodelextension"
namespaceUri="org.my.extension.mymmodelextension"
version="1"
xsi:schemaLocation="http://org.teiid.modelExtension/2011
modelExtension.xsd"
xmlns="http://org.teiid.modelExtension/2011">
<p:description>This is my model extension</p:description>
<p:extendedMetaclass
name="com.metamatrix.metamodels.relational.impl.BaseTableImpl">
  <p:property advanced="false" index="true" masked="false"
name="copyable" required="false" type="boolean">
    <p:description locale="en_US">Indicates if table can be
copied</p:description>
    <p:display locale="en_US">Copyable</p:display>
  </p:property>
</p:extendedMetaclass>
</modelExtension>
```

MED レジストリーには、**Teiid Designer** によって使用される MED が保存されます。拡張エディターで **.mxd** ファイルを開くと MED ファイルを編集できます。

### 5.3. モデル拡張定義レジストリー (MED レジストリー)

MED レジストリーは、ワークスペースに登録されたすべての MED をトラッキングします。モデルの拡張には登録された MED のみを使用できます。レジストリーに保存される MED には 2 つの種類があります。

- ※ ビルトイン MED: これらの MED は Teiid Designer のインストール中に登録されます。ユーザーは更新や登録解除することはできません。
- ※ ユーザー定義 MED: これらの MED はユーザーによって作成されます。ユーザーは更新、登録、および登録解除することができます。

MED レジストリーの状態は永続化され、新しいセッションが開始されるたびにリストアされます。

## 第6章 サーバー管理

### 6.1. JBoss Data Virtualization Server のターゲット

#### 6.1.1. JBoss Data Virtualization Server のターゲット

Teiid Designer のモデリングでは **JBoss Data Virtualization Server** が目的となります。モデルがデプロイされるサーバーの正しいバージョンを定義する必要があります。これには、サーバーバージョン設定を設定するか、サーバービューで **JBoss Data Virtualization Server** を定義します。

#### 6.1.2. サーバーバージョン設定

デフォルトのサーバーバージョン設定では、実際に **Teiid Designer** で **JBoss Data Virtualization Server** を定義しなくても対象のサーバーバージョンが変更されます。可能な値のリストは、アプリケーションインストールされている Teiid ランタイムクライアントプラグインに応じて決定されます。

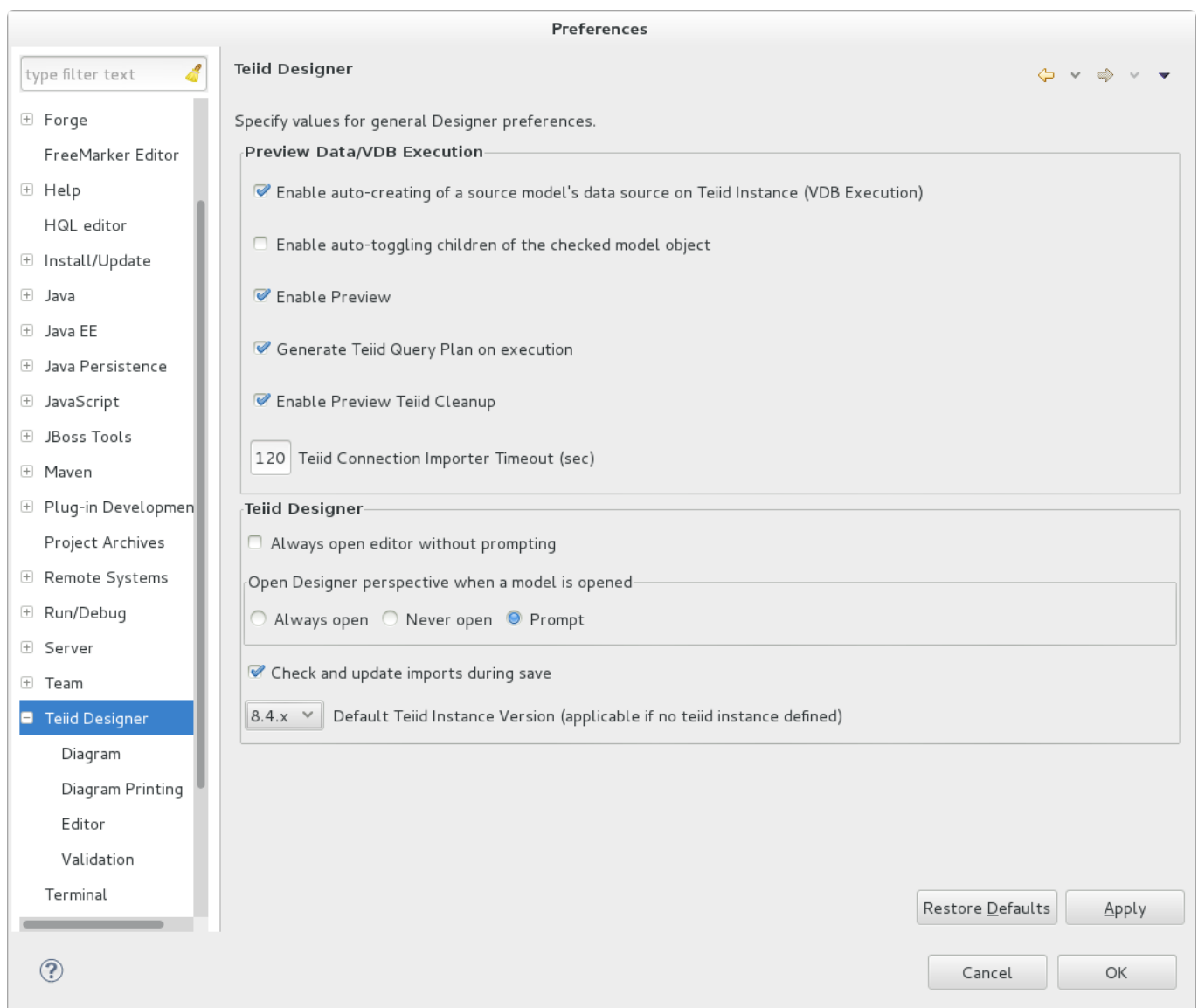


図6.1 デフォルトのサーバーバージョン設定

#### 6.1.3. JBoss Data Virtualization Server の定義



JBoss Data Virtualization Server を定義すると、モデルのデプロイメントをプレビューおよびテストできます。定義できるサーバーの数に制限はありません。しかし、サーバービューセクションのコンテキストメニューのアクションを使用しないと、常にプレビューとデプロイメントにデフォルトのサーバーが使用されます。

ガイドビューは以下の **JBoss Data Virtualization Server** アクションを提供します。

- ✧ 新しい JBoss サーバーの設定
- ✧ JBoss/Teiid インスタンスプロパティの編集
- ✧ デフォルトの JBoss/Teiid インスタンスの設定
- ✧ Teiid インスタンスのリフレッシュ

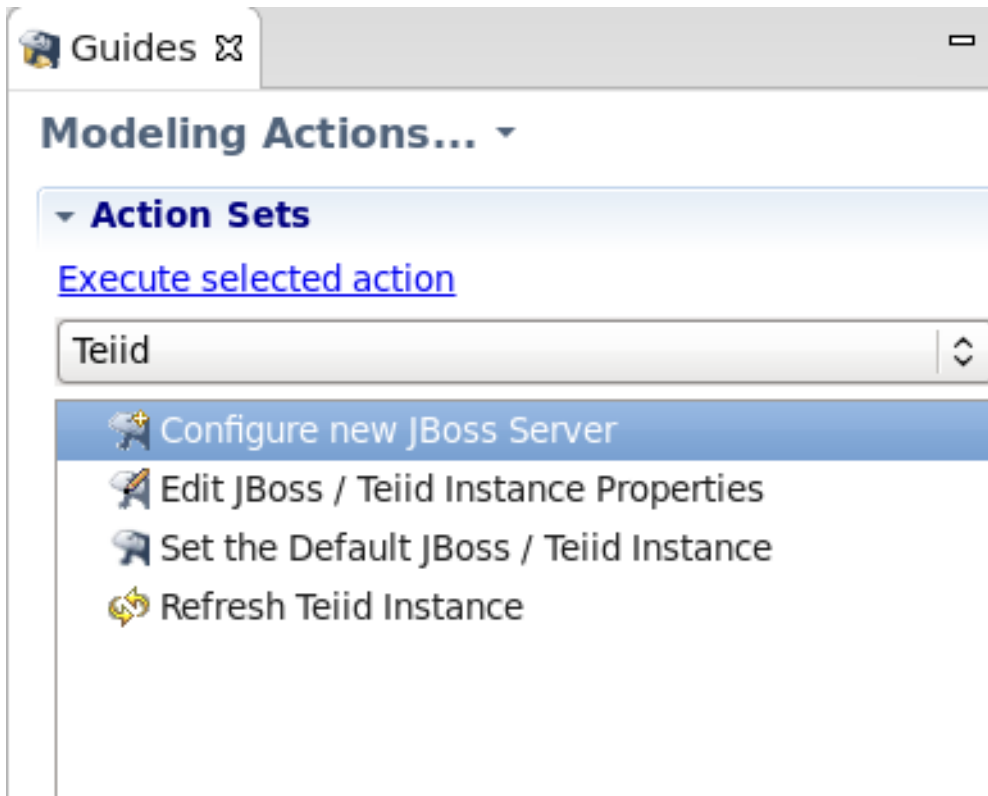


図6.2 ガイドビューの Teiid サーバーカテゴリー

**Configure New JBoss Server** アクションは、「サーバーの設定」の項に説明のあるウィザードと、サーバービューで JBoss Data Virtualization インスタンスとその親 JBoss サーバーの両方を作成するステップを表示します。

サーバービューで複数の JBoss Data Virtualization Server を定義する場合は、**Set the Default JBoss/Teiid Instance** のアクションでデフォルトのサーバーを適切に変更できます。サーバービューで現在選択された JBoss Data Virtualization サーバーはデフォルトのサーバーとして認識されます。しかし、何も選択されていないとダイアログが表示され、デフォルトのサーバーを選択するよう要求されます。

### 注記

定義された JBoss Data Virtualization サーバーのバージョンは、Preference ウィンドウで選択されたサーバーバージョンよりも常に優先されます。

### 6.1.4. サーババージョン状態パネル

サーババージョン設定の変更またはサーバの定義が行われると、デフォルトのサーバ状態パネルでサーバの対象バージョンが更新されます。これは対象となる現在のサーババージョンと、プレビューまたはデプロイを実行するために使用されるサーバを常に反映します。

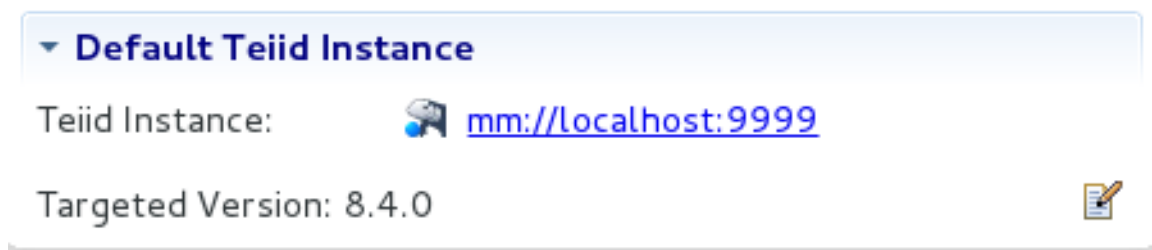


図6.3 デフォルトのサーバ状態パネル

## 6.2. サーバの設定

Teiid Designer は JBoss Developer Studio のコンポーネントとしてインストールされるため、JBoss Data Virtualization Server へ接続するには親の JBoss Server の設定が必要になります。この設定を行うにはサーバビューを使用します。Teiid Designer パースペクティブの一部として表示されるサーバビューセクションを確認してください。

これまで作成されたサーバがない場合は、サーバビューに新しいサーバハイパーリンクが表示されます。新しい JBoss Server 設定を作成するには、ハイパーリンクをクリックします。

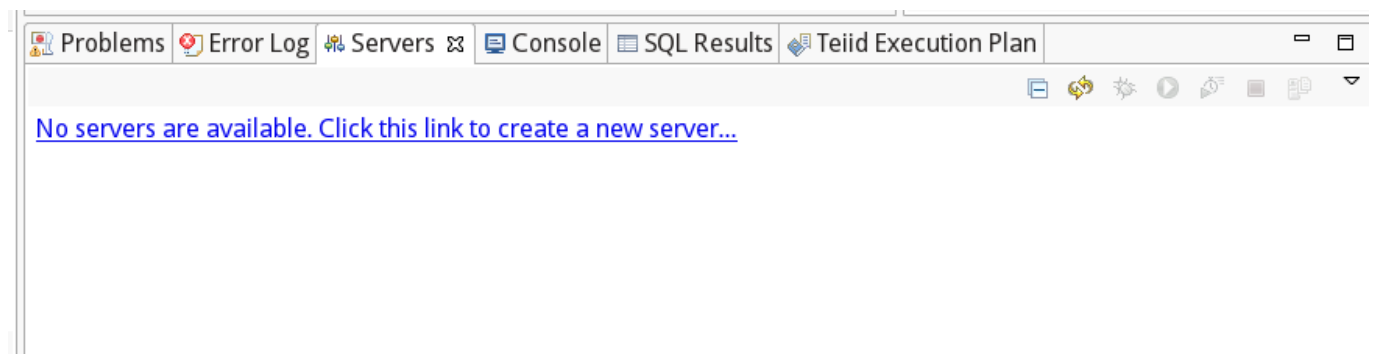


図6.4 作成されたサーバのないサーバビュー

ウィザードに従って、ランタイムの場所、ホスト名、外部から管理されるかどうかなど、JBoss Server の詳細を設定します。最後のプロパティは、サーバが IDE 内でインスタンス化されるか、または独立してインストールおよび開始されるかを決定します。後者の場合、サーバビューは独立したサーバへの接続を仮定します。


**New Server**

**Define a New Server**  
Choose the type of server to create

[Download additional server adapters](#)

Select the server type:

type filter text

 JBoss Enterprise Application Platform 6.1+

JBoss Enterprise Application Platform (EAP) 6.1+

Server's host name: localhost

Server name: jboss-eap-6.1 Runtime Server (1)

Server runtime environment: jboss-eap-6.1 Runtime [Add...](#)

[Configure runtime environments...](#)

? < Back Next > Cancel Finish

図6.5 新しいサーバーの設定

サーバーは JBoss Data Virtualization Server とインストールされています。緑色の開始ボタンをクリックすると、Teiid Designer がサーバーに接続され、JBoss Data Virtualization Server の設定が表示されます。

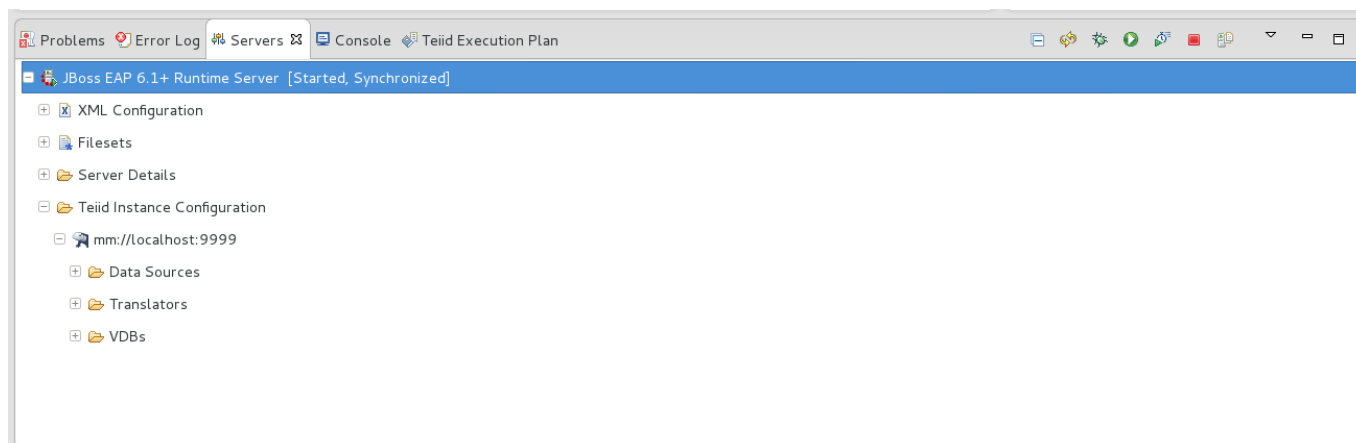


図6.6 単一サーバーのサーバービュー

JBoss Tools には JBoss Data Virtualization Server を設定するためのエディターがあります。また、Teiid Designer にはこのエディターの追加タブがあり、JBoss Data Virtualization Server の設定が表示されます。ほとんどの設定は親の JBoss Server によって決定されるため、いくつかのオプションのみが変更可能です。このエディターを表示するには、JBoss Server ツリーのノードを1つダブルクリックします。

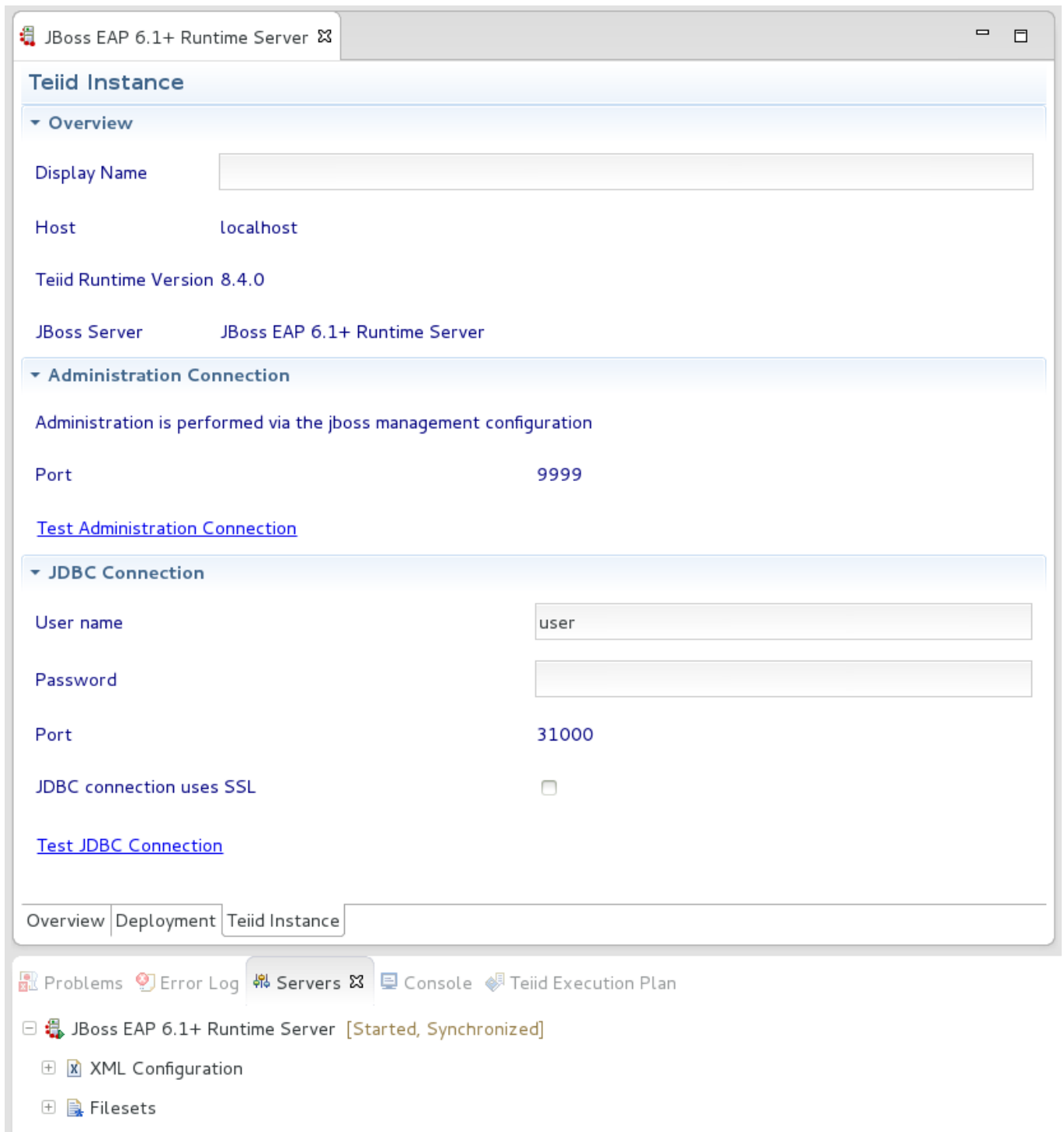


図6.7 Teiid サーバー設定用のエディター

### 6.3. JBDS をリモートの Red Hat JBoss Data Virtualization Server へ接続する

1. File - New - Other - Server を選択します。
2. Red Hat JBoss Middleware -> JBoss Enterprise Application Platform 6.x+ を選択します。
3. ホストをリモートサーバーのホスト名 (myserver01.labs.mycompany.com など) に設定し、Next をクリックします。
4. Remote を選択してから Management Operations を選択し、Server is externally managed にチェックを入れ、Assign Runtime のチェックを外してから Next をクリックします。

5. New Host をクリックして SSH Only を選択してから、Next をクリックします。
6. Hostname は事前記入のようですが、そうでない場合は自分のホスト名 (つまり、myserver01.labs.mycompany.com) をペーストして、接続名を設定してから、Finish をクリックします。
7. 再度 Finish をクリックします (Remote Server Home エントリーは記入する必要はありません)。
8. サーバービューでサーバーをダブルクリックして、admin のユーザー名/パスワードを設定し、保存します。



### 注記

ローカルマシンにユーザーを設定する必要はありません。

## 第7章 Teiid Designer の例

### 7.1. Teiid Designer の例

ここでは、一般的な作業の例をいくつか実践します。これらの例では、一般的な設計作業で活用できる、Teiid Designer に組み込まれている機能を簡単に紹介します。以下の概念を紹介します。

#### ✦ Guides

Guides ビューは、多くの一般的なモデリング作業で適切な開始点と言えます。ビューにはカテゴリー化されたモデリングアクションが含まれ、一般的なタスクの Cheat Sheets へのリンクも含まれます。カテゴリー化されたモデリングアクションは、作業を完成させるために必要となるすべてのアクションをグループ化します。Teiid Designer のメニューからアクションを探さずに、ガイドビューから直接アクションを開始できます。

#### ✦ Cheat Sheets

チートシートはカテゴリー化されたアクションセットにとどまらず、一般的な作業の一部をステップごとに行います。各ステップにおいて、可能であれば前のステップで入力したデータがプロセス全体で引き継がれます。

Guides と Cheat Sheets が動作していることを確認したら、さまざまな概念やアクションを詳細に説明する後続の章を参照してください。

### 7.2. Guides ビュー

この項では、簡単な例を用いて Guides ビューを紹介します。この例では、Model JDBC Source Action Set に従います。アクションは以下の順になります。

1. Teiid モデルプロジェクトの定義
2. JDBC 接続の作成
3. JDBC データソースのソースモデルの作成
4. データのプレビュー
5. VDB の定義
6. VDB の実行

アクション名を見ればどのようなアクションかが分かります。ワークスペースで新しいモデルプロジェクトを作成し、MySQL データベースへ接続プロパティを定義します。その後、データベースへ接続してメタデータをインポートし、**Teiid Designer** でソースモデルを作成します。次に、データベースの内容のプレビューを表示します。最後に、VDB を定義した後に実行している **JBoss Data Virtualization Server** ヘドプロイし、実行します。

### 7.3. JDBC ソースのモデリング

この項では、**Guide View** アクションセットを使用して JDBC ソースをモデリングする方法を説明します。この例では、MySQL データベースへ接続しますが、同じプロセスを使用して他のサポートされるデータベースに接続できます。

1. Guides ビューを開く

Teiid Designer の Guides ビューを開くには、メインメニューで **Window > Show View > Other...** の順にクリックし、ダイアログの **Teiid Designer > Guides** ビューをクリックします。

以下は **Model JDBC Source** アクションセットが選択された Guides ビューになります。

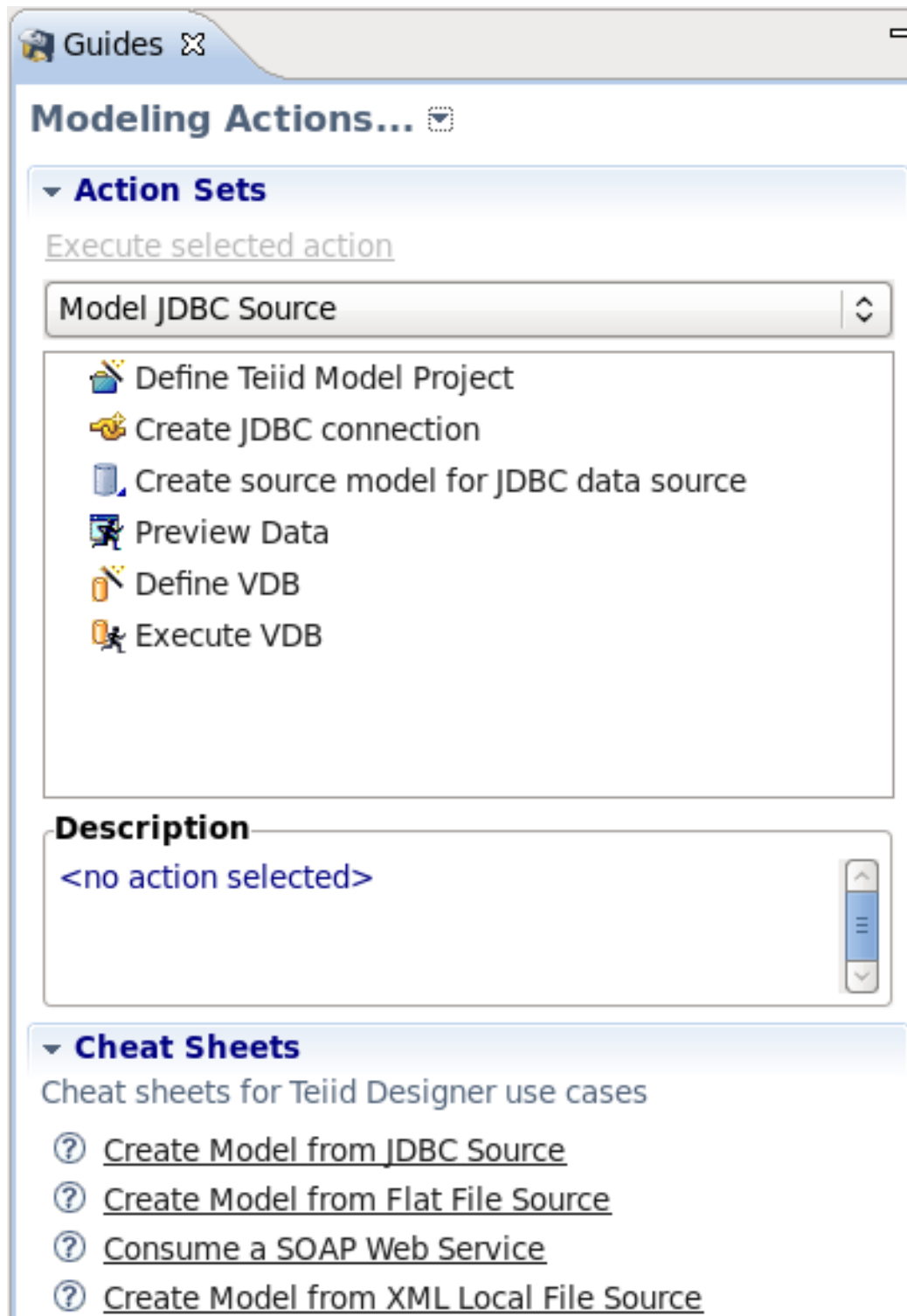


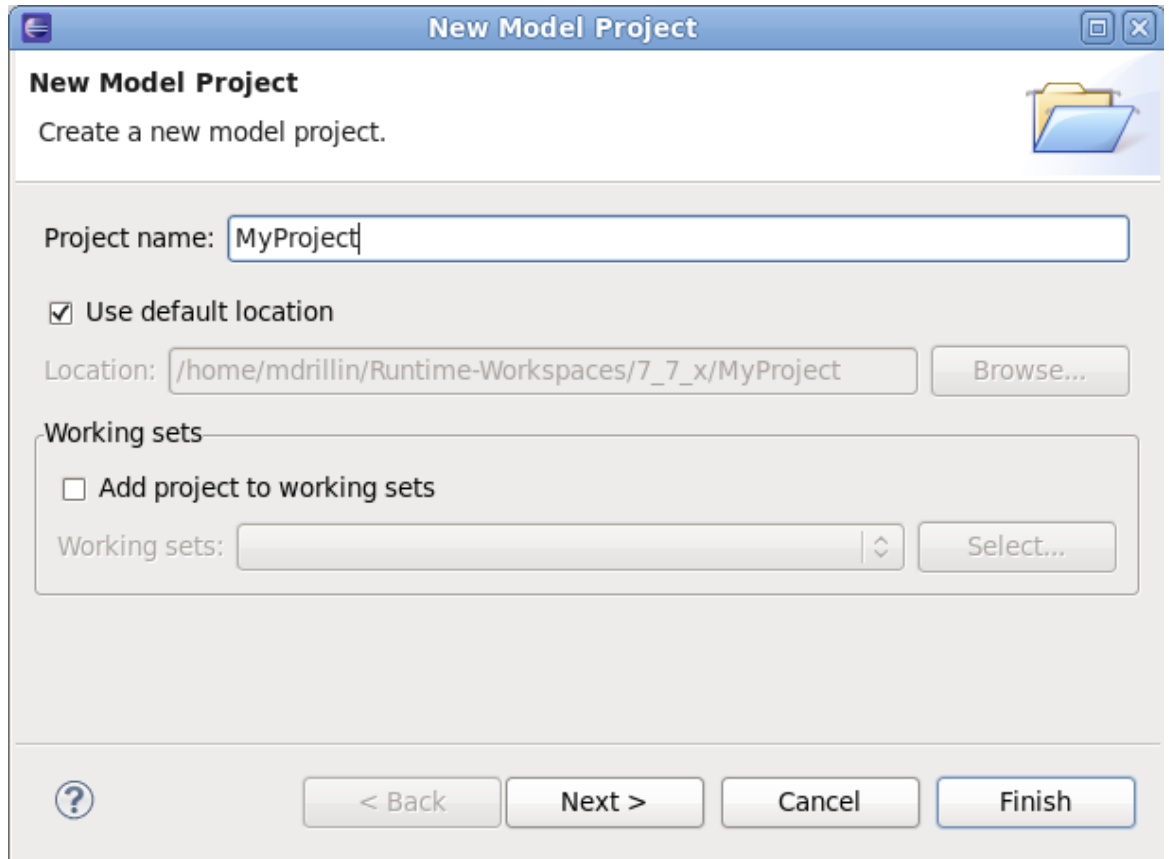
図7.1 Guides ビュー

## 2. Teiid モデルプロジェクトの定義



- a. **Define Teiid Model Project** アクションは **New Model Project** ウィザードを開始します。Action Set リストでアクションをダブルクリックします (または、アクションを選択した後に **Execute selected action** をクリックします)。開始されたウィザードは次のようになります。

プロジェクト名 (例: MyProject) を Name フィールドに入力した後、**Next** をクリックします。



The screenshot shows the 'New Model Project' wizard dialog box. The title bar is 'New Model Project'. The main text is 'New Model Project' and 'Create a new model project.' with a folder icon. The 'Project name' field contains 'MyProject'. The 'Use default location' checkbox is checked. The 'Location' field shows the path '/home/mdrillin/Runtime-Workspaces/7\_7\_x/MyProject' with a 'Browse...' button. The 'Working sets' section has an unchecked 'Add project to working sets' checkbox and a 'Working sets' dropdown menu with a 'Select...' button. At the bottom, there are buttons for '< Back', 'Next >', 'Cancel', and 'Finish', along with a help icon.

図7.2 New Model Project ウィザード

- b. ウィザードの次のページは以下ようになります。

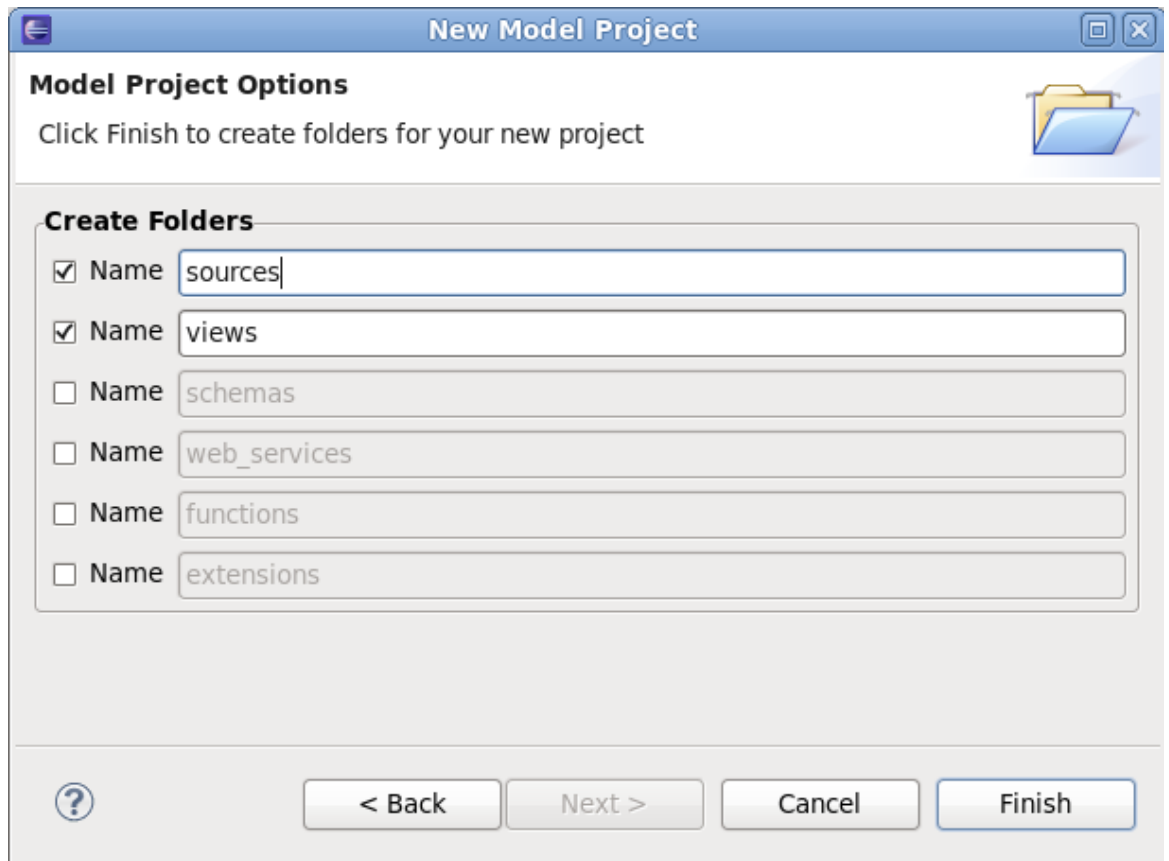


図7.3 新規プロジェクトのフォルダー

- c. **Create Folders** 下の **schemas** および **web\_services** を消去します。これらはこの例では必要ありません。次に、**Finish** をクリックしてウィザードを終了します。これでプロジェクトが作成されます。

Model Explorer ビューは以下のようにになります。

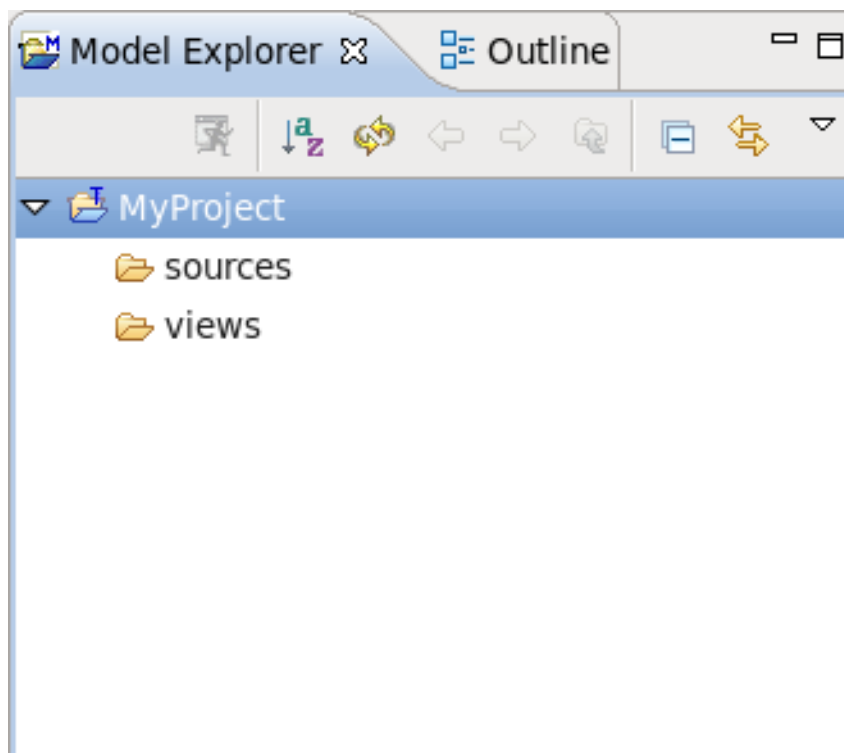


図7.4 Model Explorer

### 3. JDBC 接続の作成

**Create JDBC connection** アクションはデータベースの **Connection profile** を作成します。接続プロファイルは、データベースへ接続するとき使用されるプロパティとドライバーを定義します。Action Set リストでアクションをダブルクリックします (または選択後に **Execute selected action** をクリックします)。下図のウィザードが起動します。

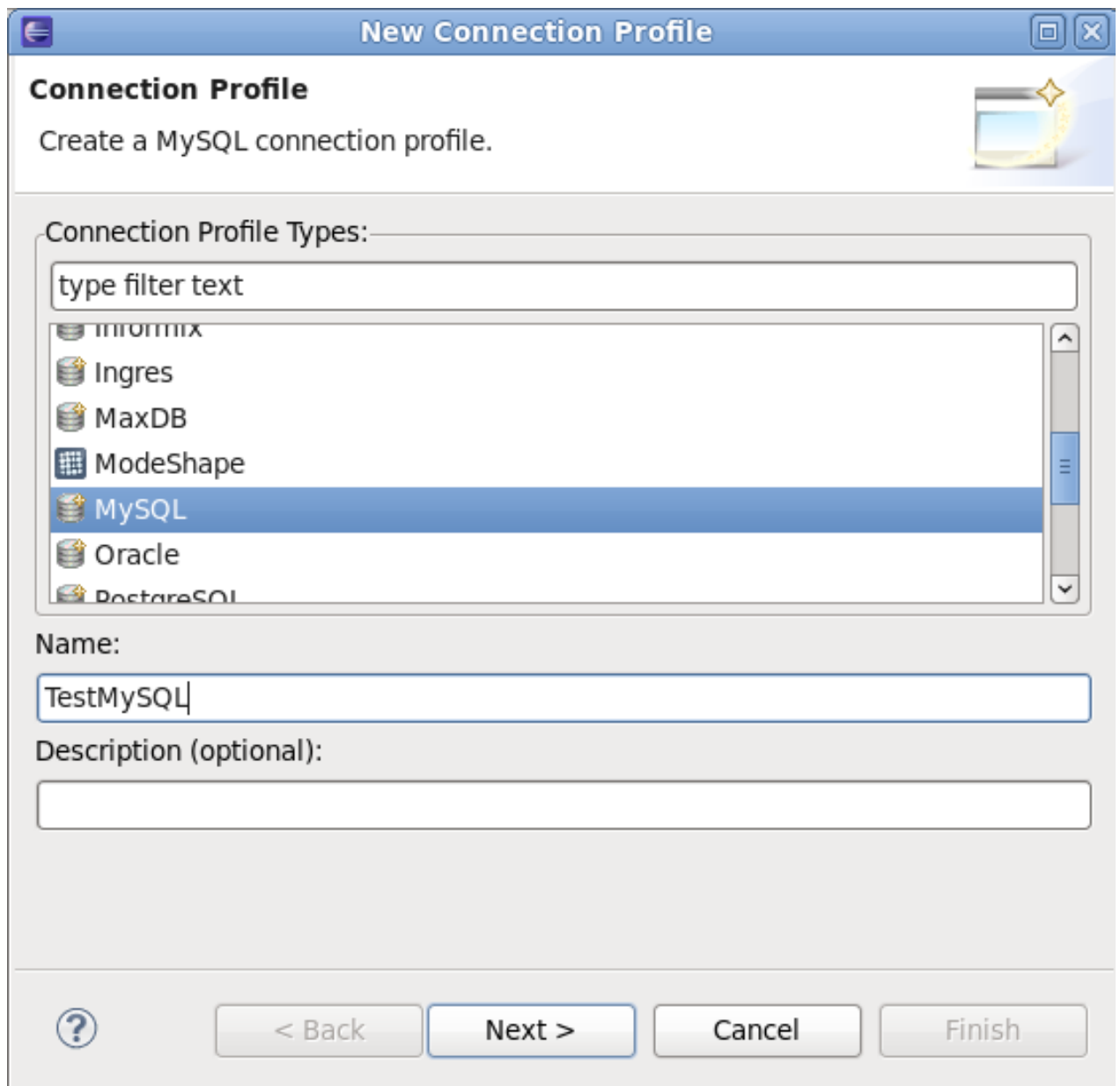


図7.5 接続プロファイル名およびタイプ

接続するデータベースのタイプを選択し (例: MySQL)、接続プロファイルの名前 (例: TestMySQL) を入力します。その後、**Next** をクリックします。

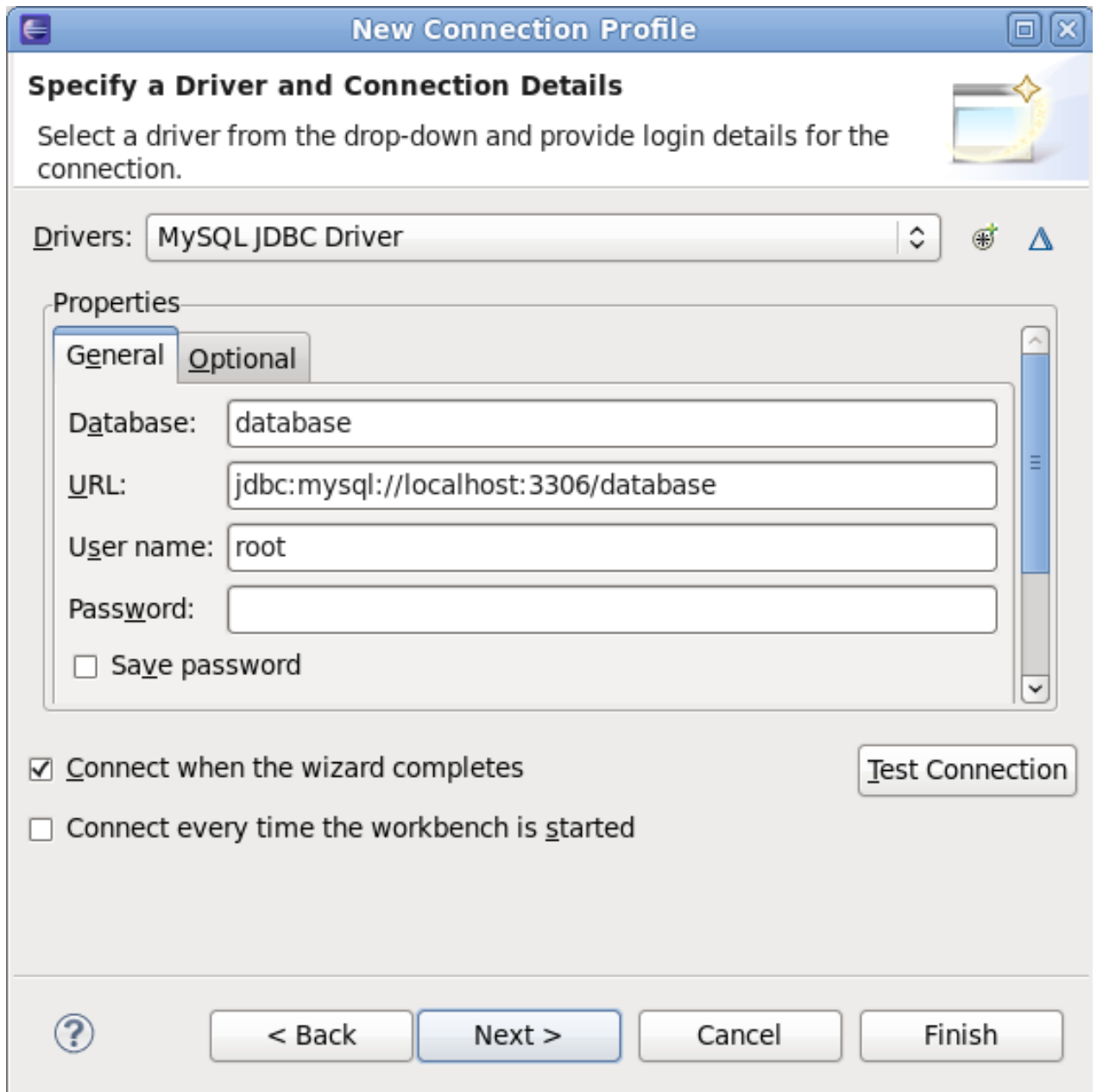


図7.6 接続プロファイルプロパティ

ドライバーを選択し、データベースのログインプロパティを入力します。**Next** をクリックし、プロファイルの作成を完了します。

#### 4. JDBC データソースのソースモデルの作成

- a. データベースからメタデータをインポートして Teiid ソースモデルを作成するため、**Create source model for JDBC data source** アクションは作成した **Connection profile** を利用するようになります。Action Set リストでアクションをクリックします (または選択後に **Execute selected action** をクリックします)。下図のウィザードが起動します。

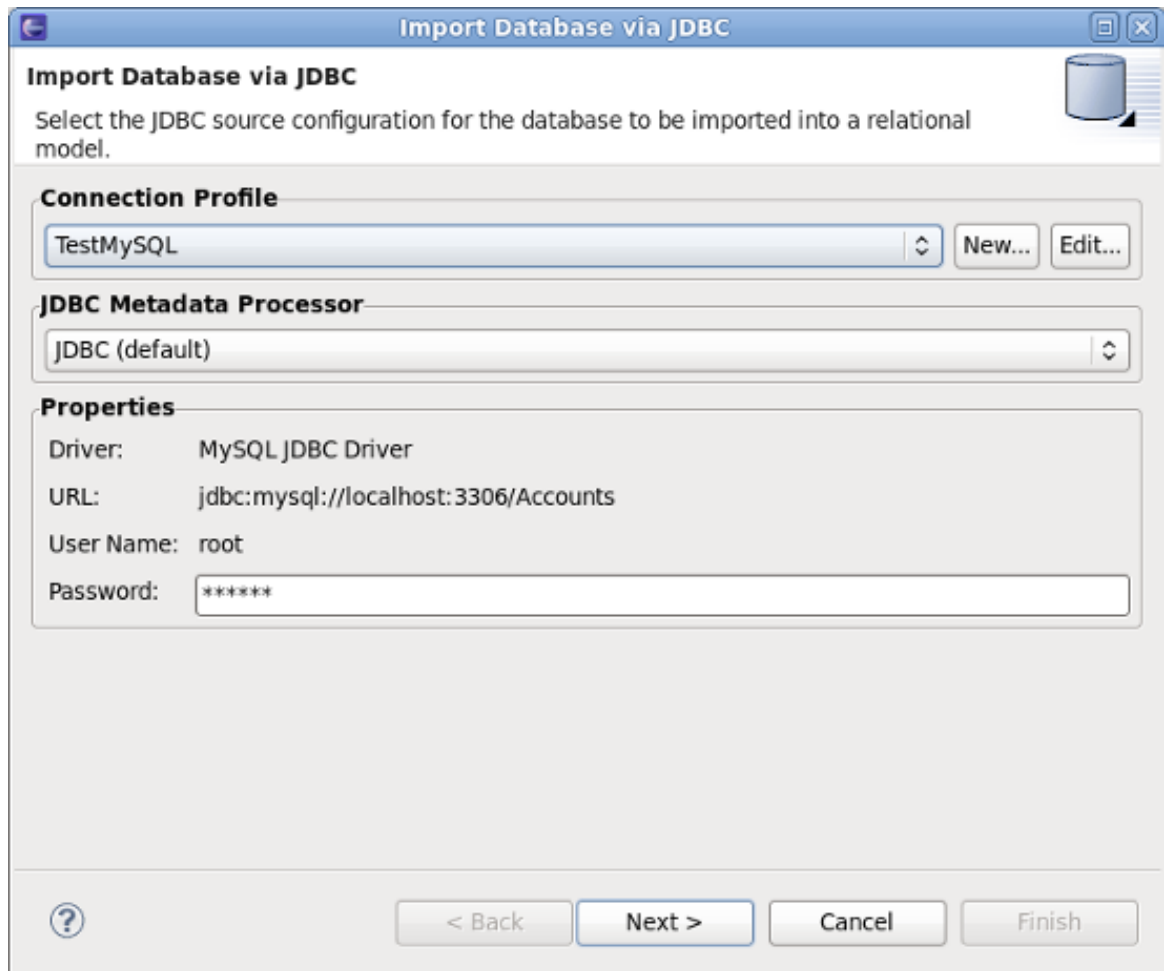


図7.7 接続プロファイルの選択

- b. このページで、前の手順で作成した **TestMySQL** 接続プロファイルを選択します。**Next** をクリックします。

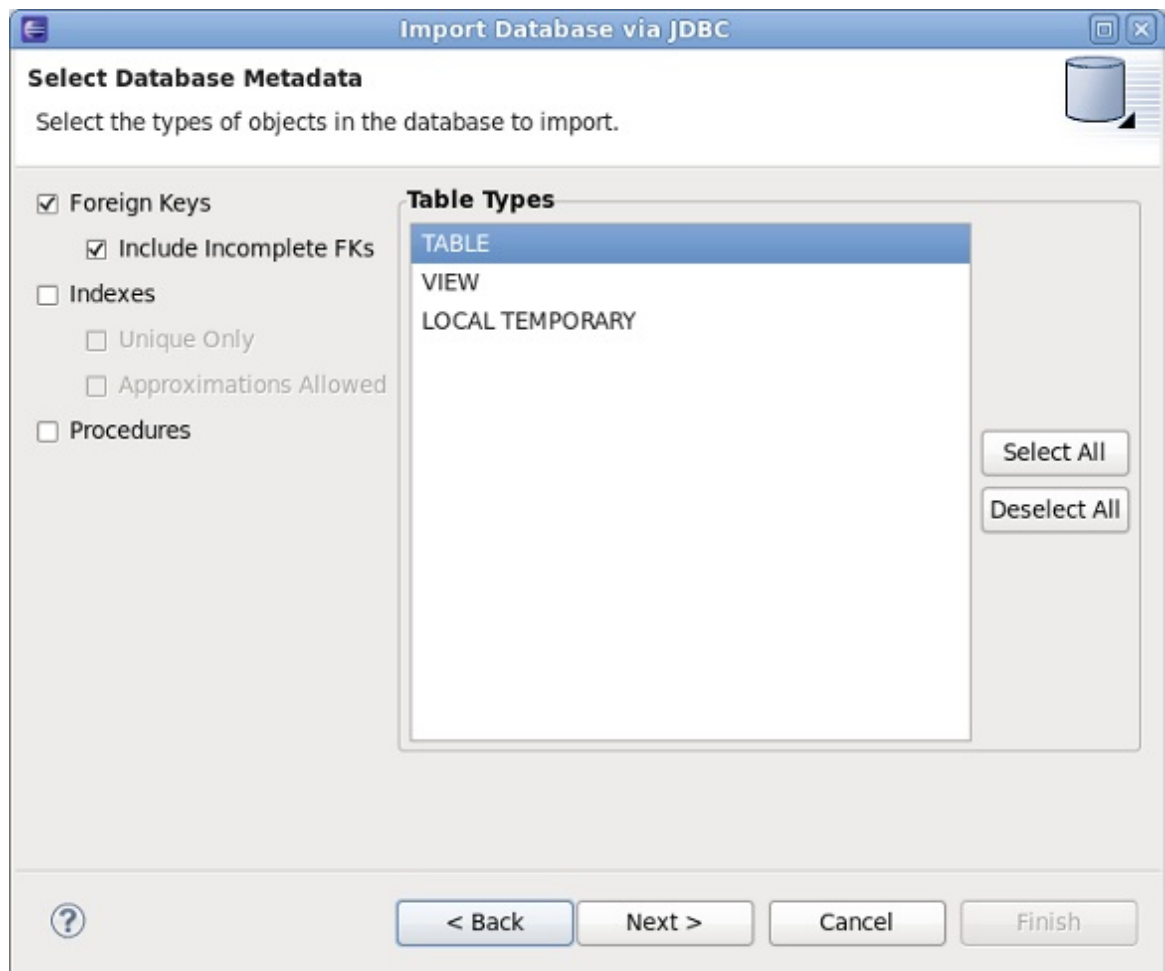


図7.8 データベースメタデータの選択

- c. このページで、インポートしたいデータベースメタデータを選択します。選択したら **Next** をクリックします。

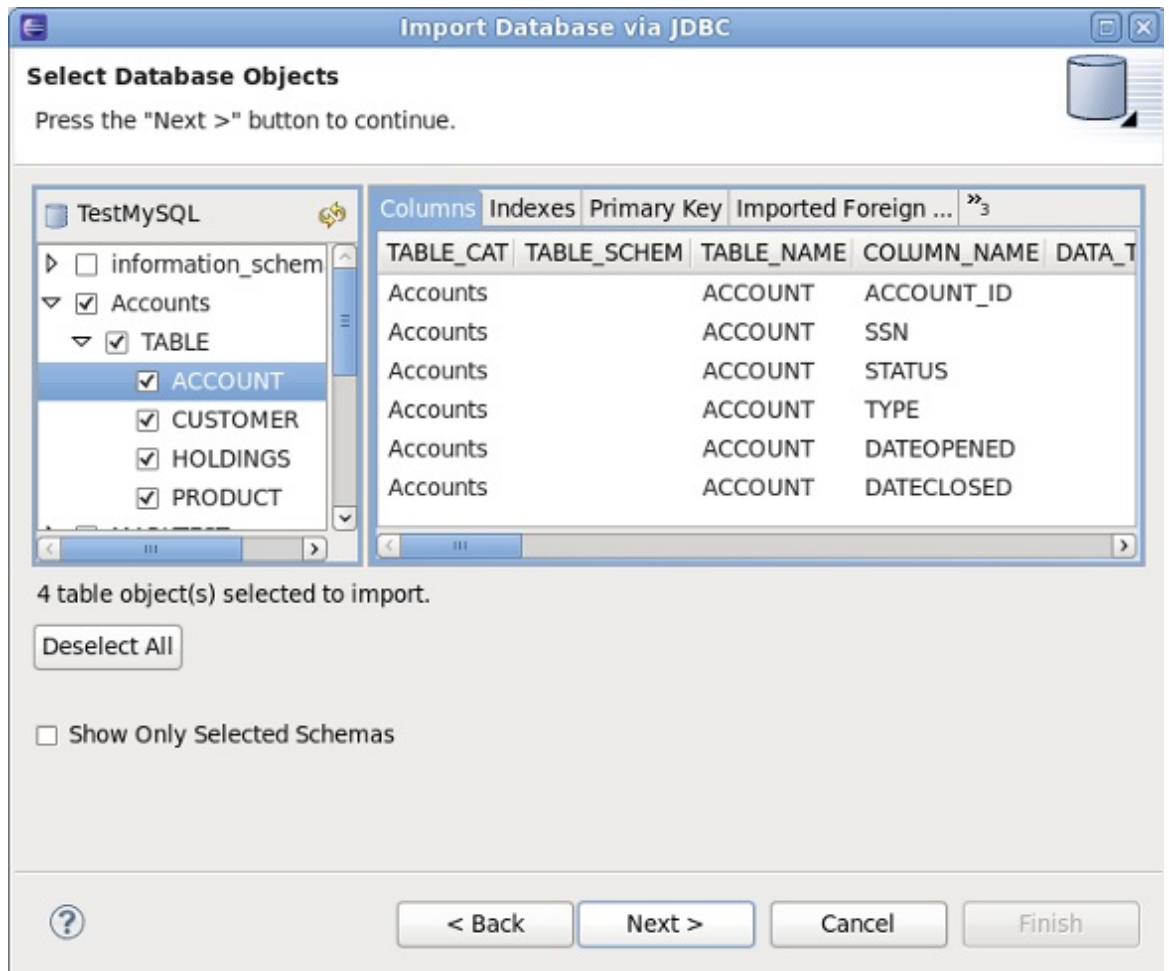


図7.9 データベースオブジェクトの選択

- d. このページで、データベースからインポートしたい特定のオブジェクトを選択します。選択したら **Next** をクリックします。

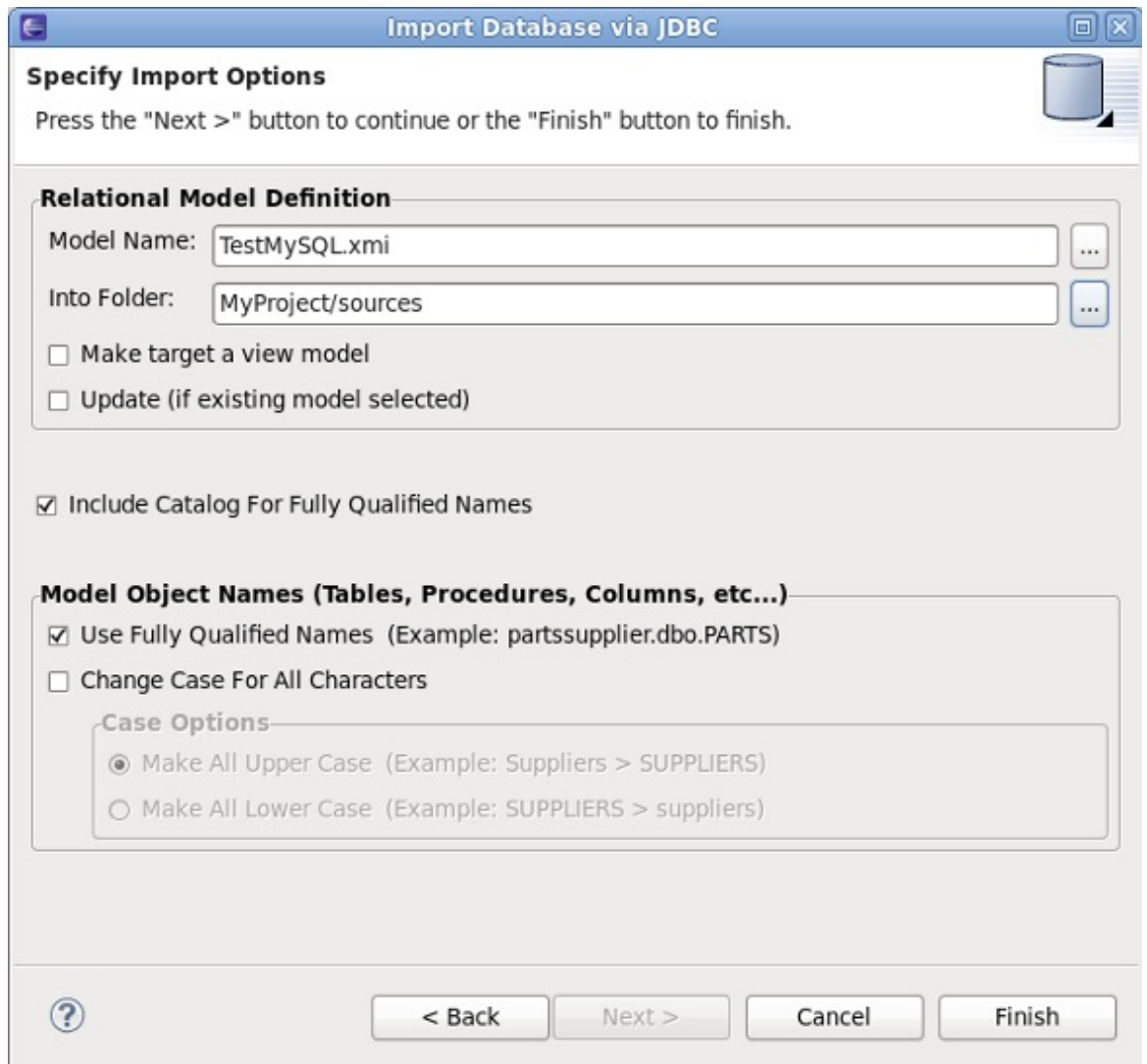


図7.10 インポートオプション

- e. 最後に、作成するモデルの名前を選択します (デフォルトは **profileName.xmi** です)。**Into Folder** フィールドは新しいモデルの目的の場所を定義します。**MyProject/sources** フォルダを選択します。選択したら **Finish** をクリックします。これでソースモデルが作成されます。Model Explorer ビューは以下ようになります。



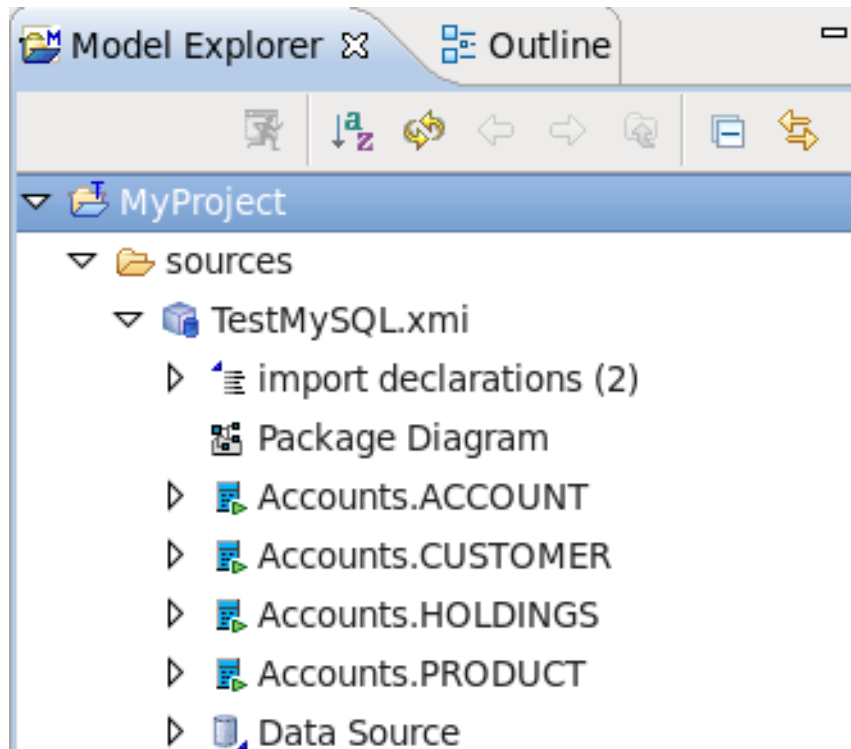


図7.11 Model Explorer

## 5. データのプレビュー

Teiid Designer のすべての実行機能 (データのプレビュー、VDB 実行) を使用するには、稼働中の JBoss Data Virtualization Server に接続する必要があります。JBoss Data Virtualization Server の接続を確立する手順は、teiid-view を参照してください。JBoss Data Virtualization Server へ接続した後、以下の手順を行うことができます。

- a. **Preview Data** アクションは、ソースからのデータ行のサンプルをプレビューできるようにします。Action Set リストでアクションをダブルクリックします (または、選択後に **Execute selected action** をクリックします)。
- b. ダイアログで、以下のようにプレビューで表示したいソーステーブルを選択します。

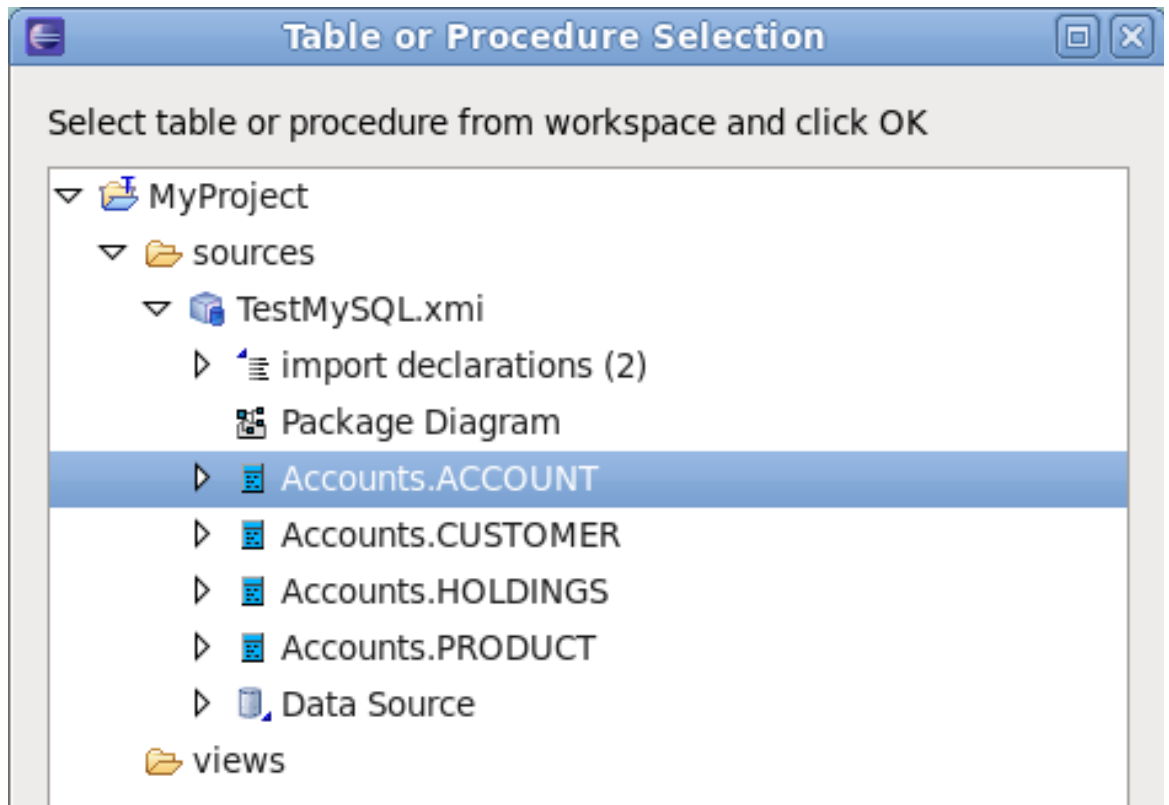


図7.12 プレビューテーブルの選択

- c. テーブルを選択したら **OK** をクリックします。プレビューの結果が表示されます。

| Status  | Operation                    | Date | Cc | ACCOUNT_ID | SSN      | STATUS   | TYPE   | DATEOPENED            | DATEC |
|---------|------------------------------|------|----|------------|----------|----------|--------|-----------------------|-------|
| Success | select * from Jun 8, 2012 Tr |      |    | 19980002   | CST01002 | Personal | Active | 1998-02-01 00:00:00.0 | 2012- |
|         |                              |      |    | 19980003   | CST01003 | Personal | Active | 1998-03-06 00:00:00.0 | 2012- |
|         |                              |      |    | 19980004   | CST01004 | Personal | Active | 1998-03-07 00:00:00.0 | 2012- |
|         |                              |      |    | 19980005   | CST01005 | Personal | Active | 1998-06-15 00:00:00.0 | 2012- |
|         |                              |      |    | 19980006   | CST01006 | Personal | Active | 1998-09-15 00:00:00.0 | 2012- |
|         |                              |      |    | 19990007   | CST01007 | Personal | Active | 1999-01-20 00:00:00.0 | 2012- |

図7.13 プレビューの結果

## 6. VDB の定義

- a. **Define VDB** アクションは、JBoss Data Virtualization Server へのデプロイメントの VDB (仮想データベース) アーティファクトを作成できるようにします。Action Set リストでアクションをダブルクリックします (または、選択後に **Execute selected action** をクリックします)。以下のダイアログが表示されます。

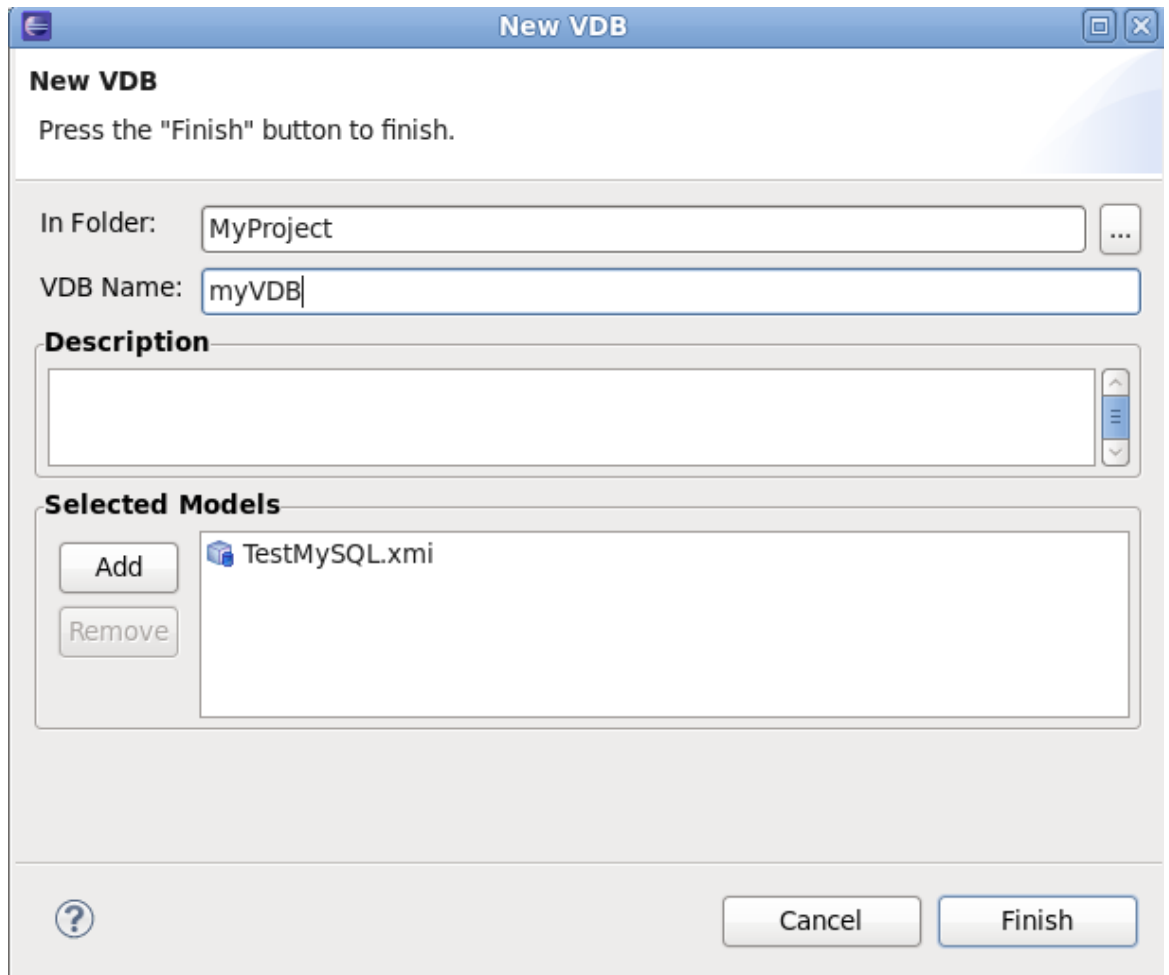
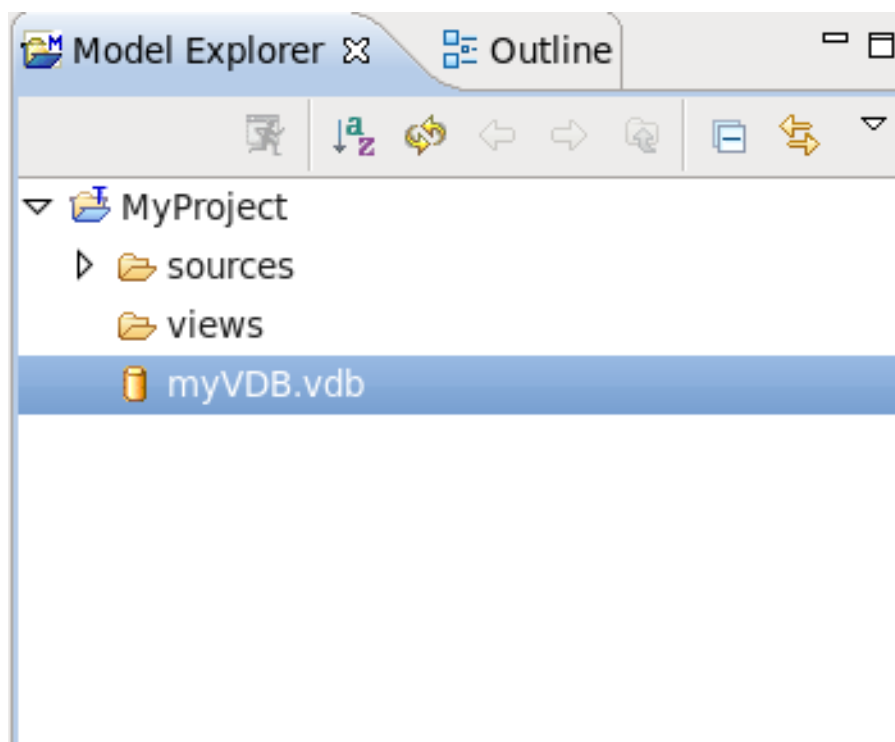


図7.14 新しい VDB

- b. ダイアログで、VDB を置く **In Folder** の場所を選択します。VDB の名前 (例: myVDB) を入力します。最後に、VDB に含まれるモデルを選択します。選択後、**Finish** をクリックします。下図のように Teiid モデルプロジェクトに VDB が作成されます。



## 図7.15 Model Explorer

### 7. VDB の実行

**Execute VDB** アクションは、VDB を実行し、その VDB に対してサンプルクエリーを実行できるようにします。Action Set リストでアクションをダブルクリックします (または、選択後に **Execute selected action** をクリックします)。ダイアログで、実行する VDB を選択した後、**OK** をクリックします。VDB がデプロイおよび実行され、**Database Development** パースペクティブに切り替わります。以下の例のように、VDB に対してクエリーを実行できるようになります。

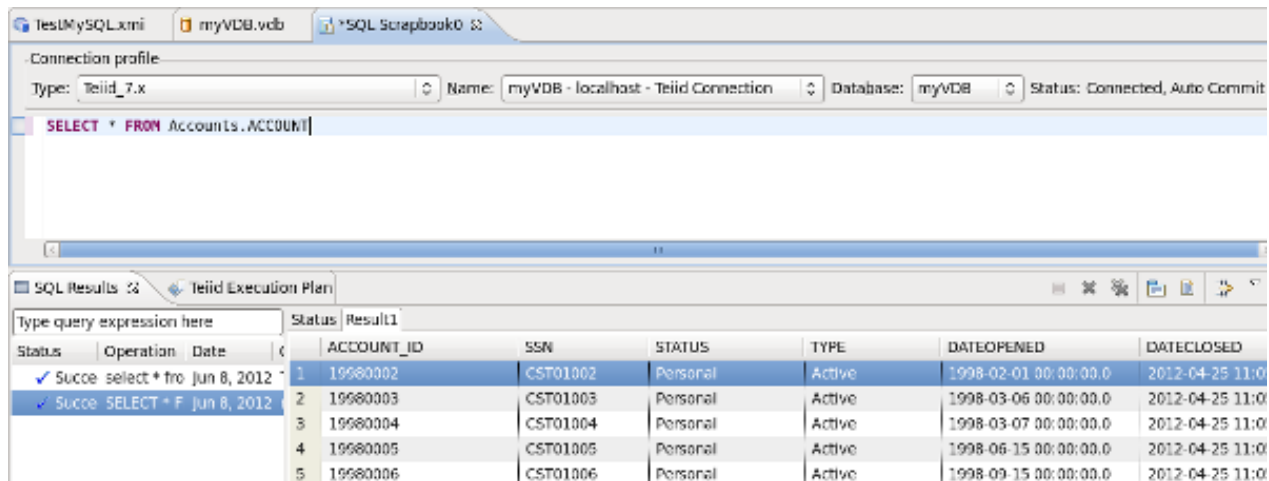


図7.16 VDB 例の実行

## 7.4. Cheat Sheets

本項では、簡単な例を用いて Cheat Sheet を紹介します。この例では、**Consume a SOAP Web Service** Cheat Sheet に従います。

## 7.5. SOAP Web サービスの消費

この項では、Cheat Sheet を使用して SOAP Web Service を消費する方法を説明します。パブリックでアクセス可能な Web サービスへの接続について実証します。このプロセスは、他の Web サービスのモデリングの例として使用できます。

### 1. Cheat Sheet を開く

**Designer** メニューから **Cheat Sheet** にアクセスできます。**Designer** メインメニューから **Window > Show View > Other...** と選択し、ダイアログで **Help > Cheat Sheets** を選択します。

**Guide View** から **Cheat Sheet** にアクセスすることもできます。以下の Guide ビューの例では、**Consume a SOAP Web Service** アクションが選択されています。

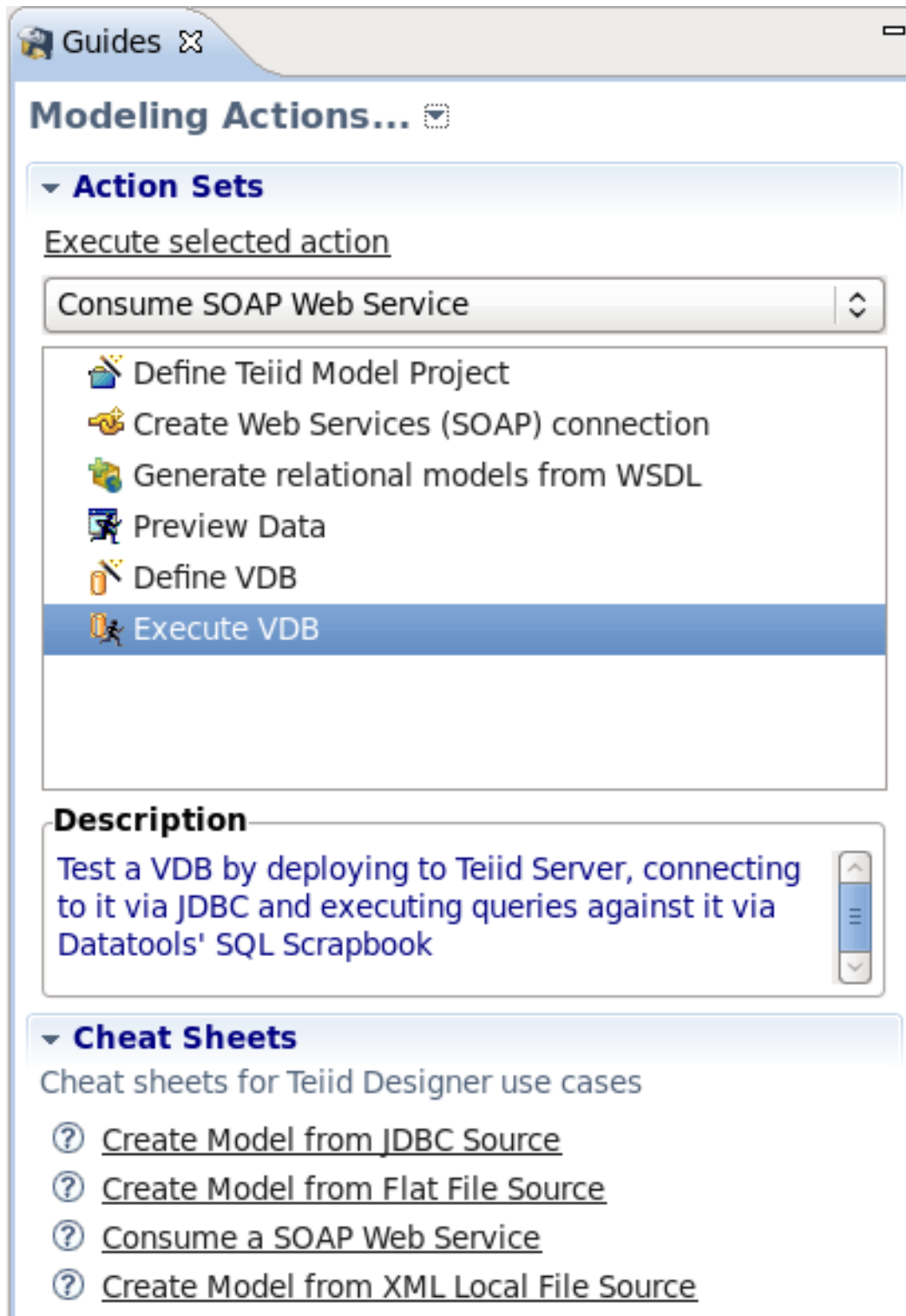


図7.17 Guides ビュー

**Guide View** から **Cheat Sheet** を開くには、**Guide View** の下部にある **Cheat Sheet** のセクションを展開し、**Consume a SOAP Web Service** リンクを選択します。

## 2. Cheat Sheet の開始

以下は **Consume a SOAP Web Service** の Cheat Sheet です。

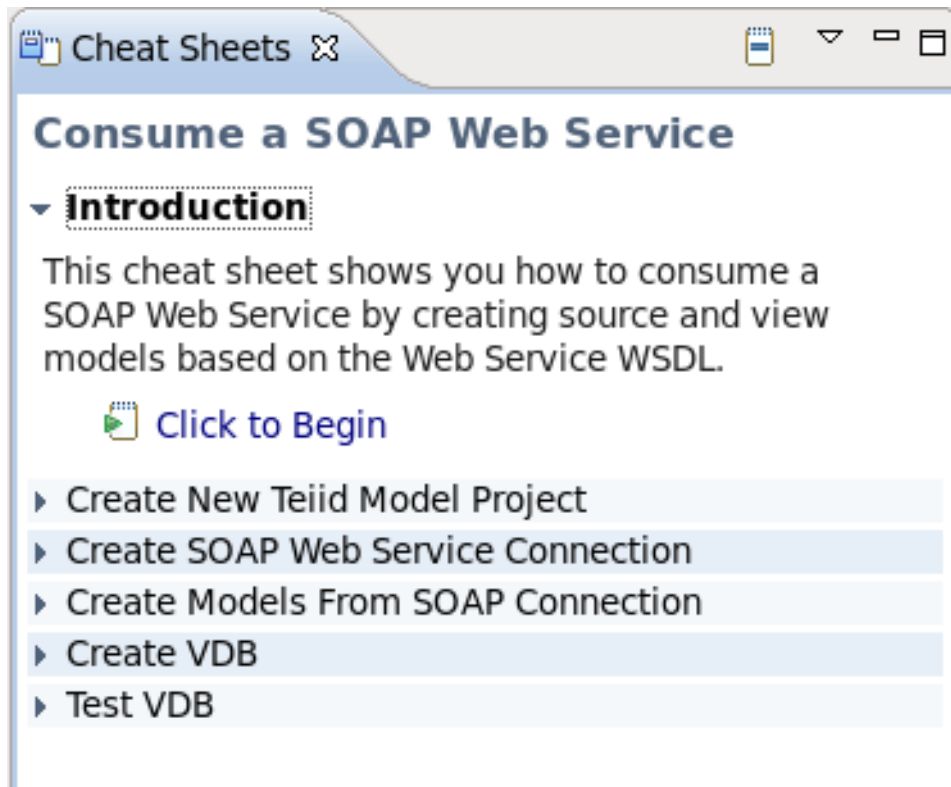


図7.18 Consume SOAP Web Service の Cheat Sheet



Cheat Sheet プロセスを開始するには、**Introduction** セクションを展開した後に **Click to Begin** を選択します。以下の **Create New Teiid Model Project** セクションが表示されます。

Cheat Sheets



## Consume a SOAP Web Service

- ✓ ▶ Introduction
- ▼ **Create New Teiid Model Project**

**Follow the steps below to create a Teiid Model Project**

Launch New Teiid Model Project Wizard  

---

- 1) Specify unique project name  
- 2) Click **Next** > twice to get to **Model Project Options** page
- 3) Check folders you wish the wizard to create in your project
- 4) Click **Finish**



- ▶ Create SOAP Web Service Connection
- ▶ Create Models From SOAP Connection
- ▶ Create VDB
- ▶ Test VDB

図7.19 モデルプロジェクトの作成



### 注記

シートの各セクションには、各ステップで何を行うかを説明する基本手順が記載されています。

Click   next to **Launch New Teiid Model Project Wizard** to launch the **New Project** wizard. Follow the wizard to create a new Model Project. For this example, we will use **SOAPPProj** for our project name. On the second page of the wizard, select the **sources** and **views** folders. Click **Finish**. The new project is created.

In the Cheat Sheet, you can advance to the next step - once the wizard has completed. Click



to advance to the next step.

### 3. Create SOAP Web Service Connection

Cheat Sheet のこのセクションには、以下のように SOAP Web Service の接続プロファイルを作成する手順が記載されています。

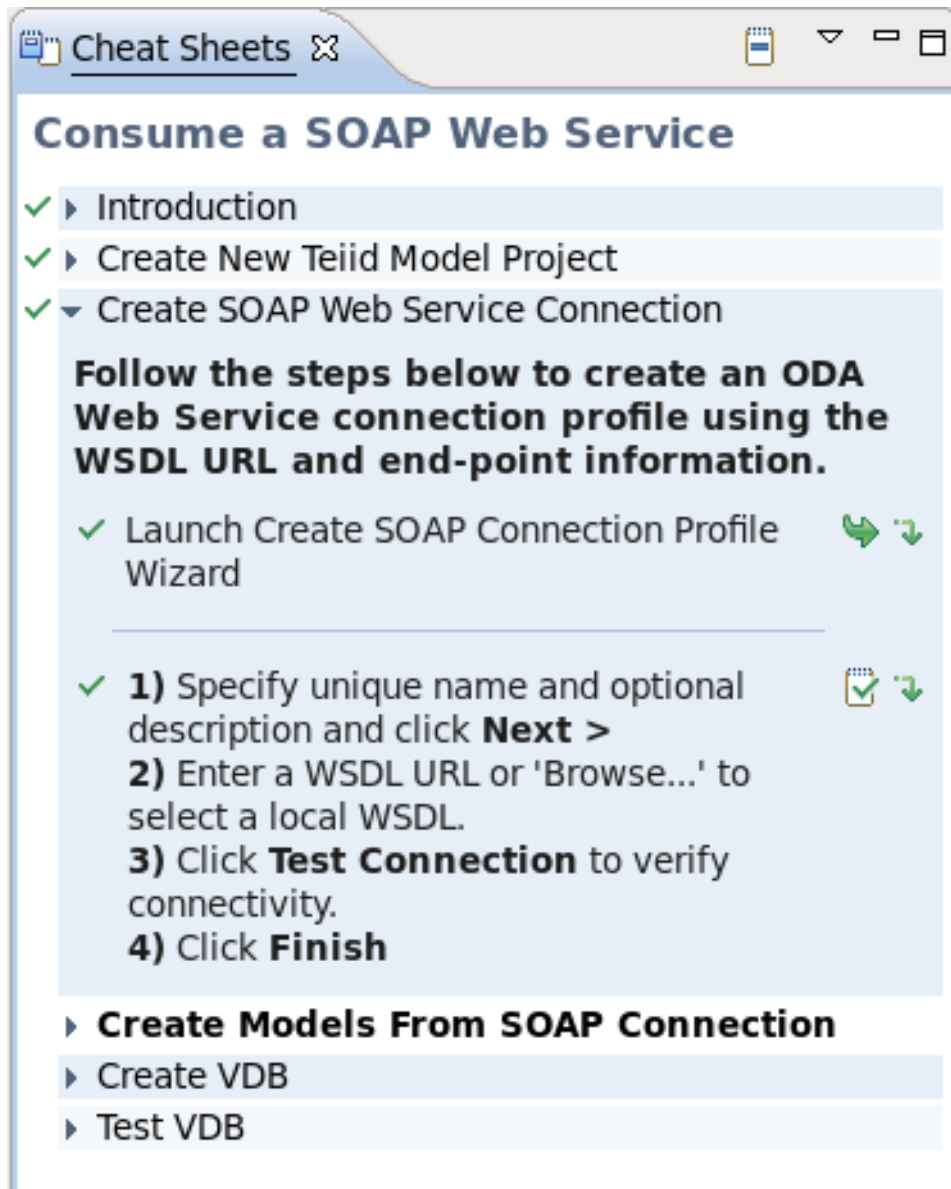



図7.20 SOAP 接続プロファイルの作成

Click  next to **Launch Create SOAP Connection Profile Wizard** to launch the wizard. The first page of the wizard is shown below:



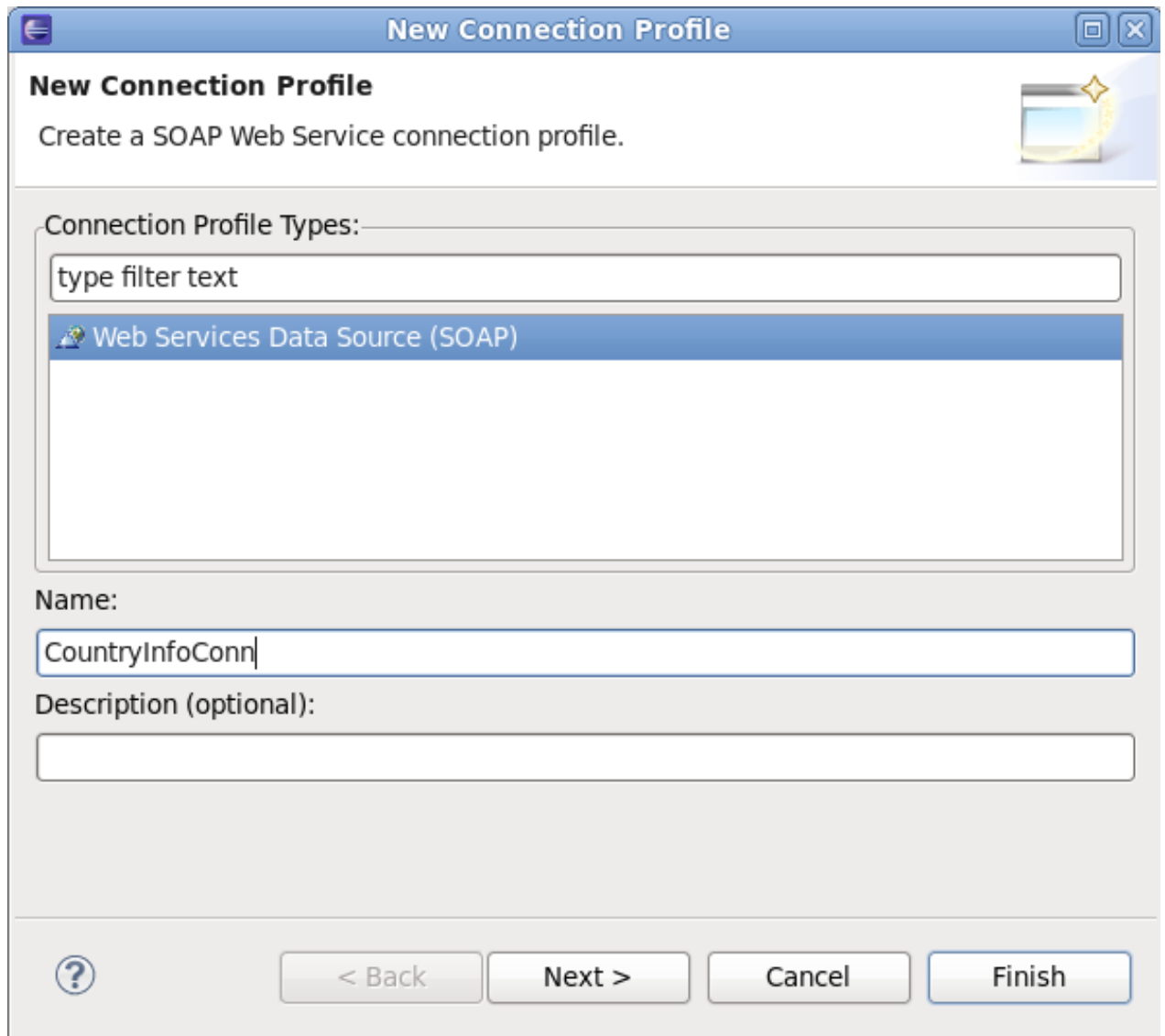


図7.21 SOAP 接続プロファイルの作成

**Web Services Data Source (SOAP)** プロファイルタイプが選択されます。プロファイル名として **CountryInfoConn** を入力し、**Next** をクリックします。以下はウィザードの次のページになります。

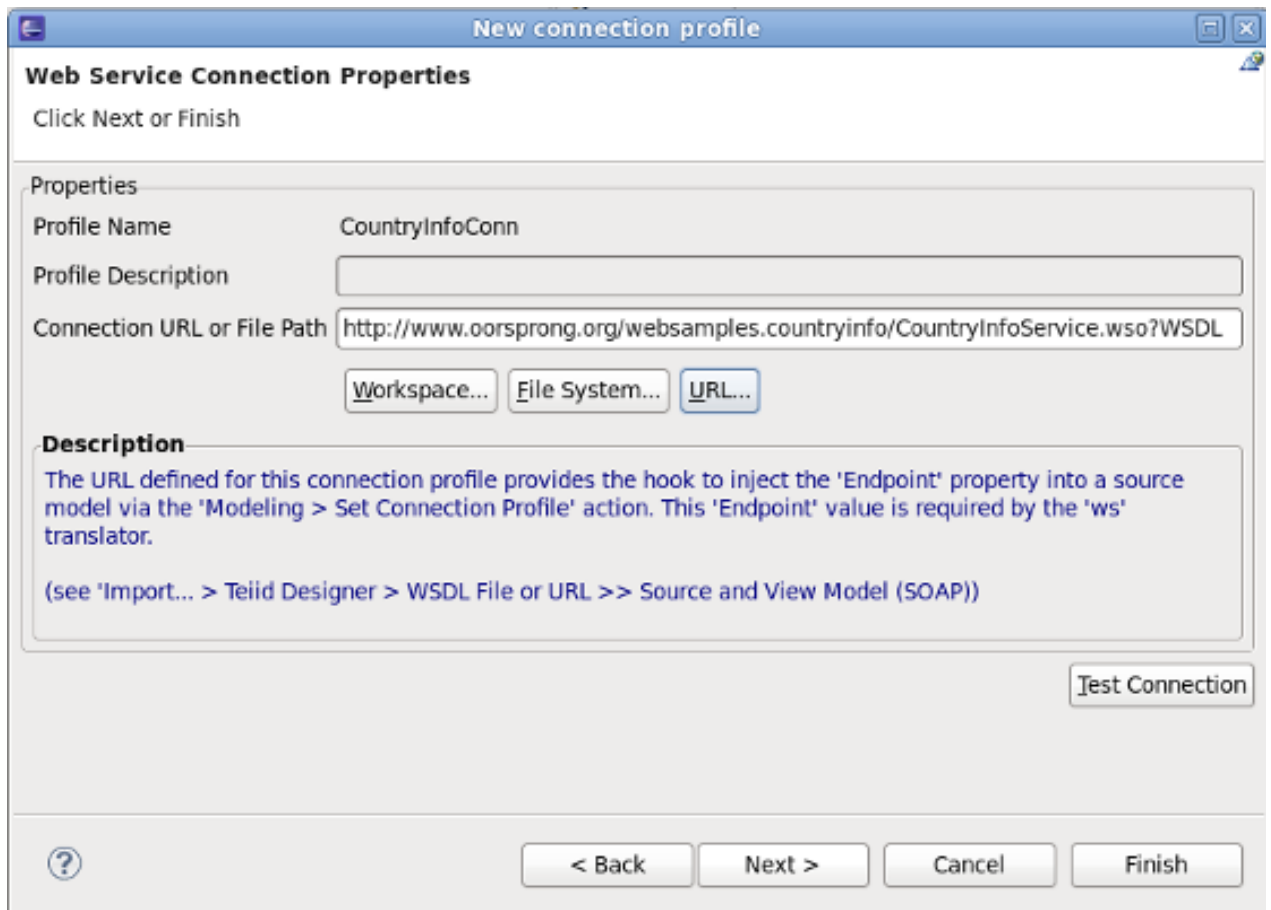


図7.22 SOAP 接続プロパティ

The connection profile properties are entered on this page. Click the **URL . . .** button, then enter the following URL: <http://www.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL>. Select **None** for **SecurityType**, then click **OK** to complete the wizard. In the Cheat Sheet, you can now continue - once the wizard has completed. Click   to advance to the next step.

#### 4. Create Models from SOAP Connection


Cheat Sheet のこのセクションには、以下のように前の手順で作成された SOAP Web Service の接続プロファイルを使用してリレーショナルモデルを作成する手順が記載されています。

Cheat Sheets



## Consume a SOAP Web Service

- ✓ ▶ Introduction
- ✓ ▶ Create New Teiid Model Project
- ↻ ▶ Create SOAP Web Service Connection
- ▼ **Create Models From SOAP Connection**

**Follow the steps below to create models using the previously defined connection profile.**

- ✓ Launch the Consume SOAP Web Service Wizard 

---

- 1) Select the connection profile, then press **Validate WSDL.**  
- 2) Select the **Port** and **Service Mode** as desired.
- 3) Select one or more operations under **Select the desired WSDL Operations.** Click **Next >**.
- 4) Select the location and name for the Source and View Models to be created. Click **Next >**.
- 5) For each operation, define the request and response XML documents:
  - Select the Operation
  - Select the **Request** tab, then select desired element(s) under schema contents and **Add** to Element Info.
  - Select the **Response** tab, then select desired element(s) under schema contents and **Add** to Column Info.
  - Select the **Wrapper Procedure** tab to view the generated procedure SQL.
- 6) When all operation(s) have been defined, click **Finish.**

- ▶ Create VDB
- ▶ Test VDB

図7.23 Create Models from SOAP Connection

Click  next to **Launch the Consume SOAP Web Service Wizard** to launch the wizard. The first page of the wizard is shown below:

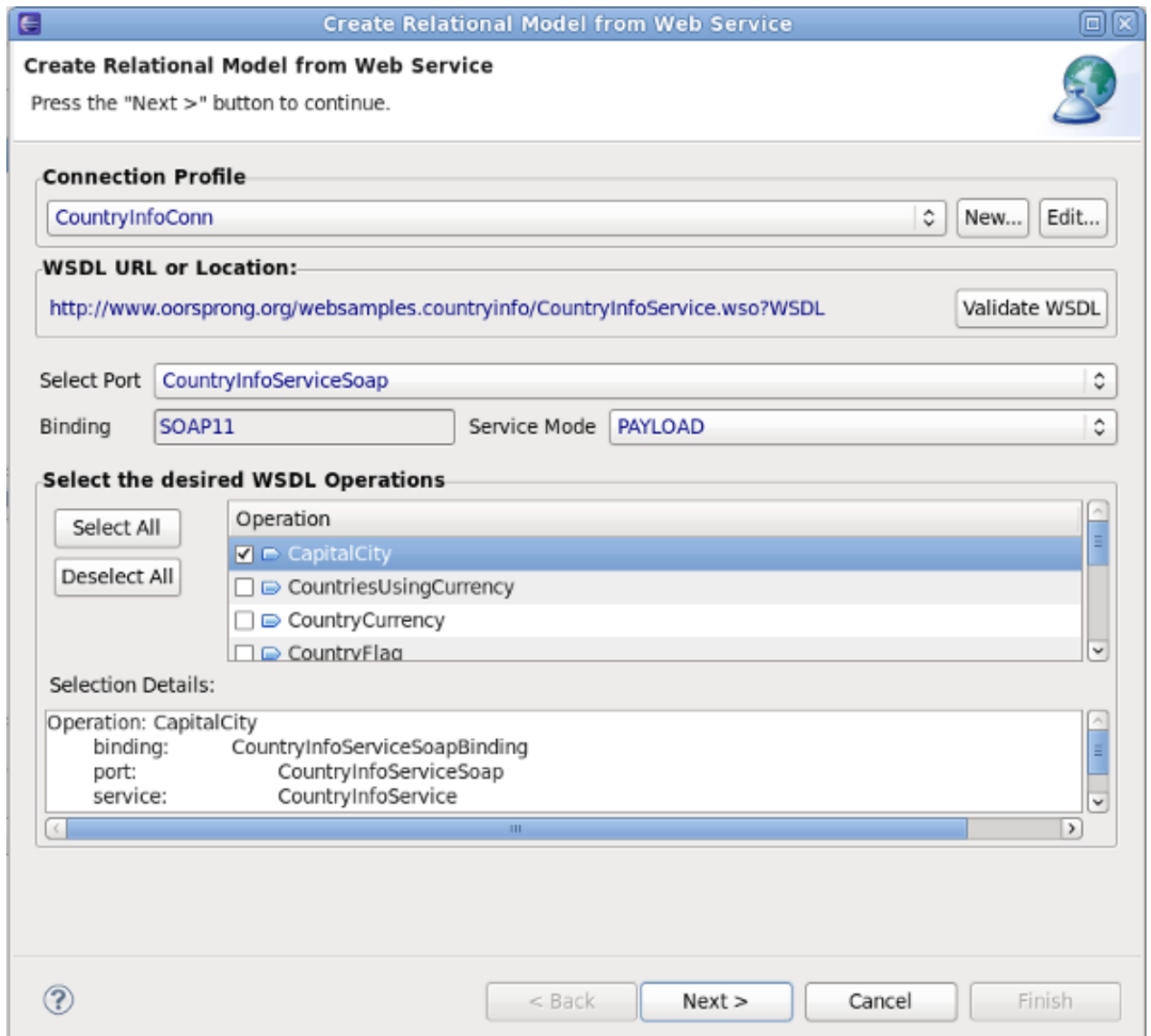


図7.24 SOAP ウィザードの消費

**Connection Profile** では、以前の手順で作成した **CountryInfoConn** プロファイルを選択します。選択すると、**Select the desired WSDL Operations** の下に利用可能な WSDL 演算が表示されます。この例では、最初の **CapitalCity** 演算のみを選択します。**Next** をクリックして、以下のように次のページに進みます。

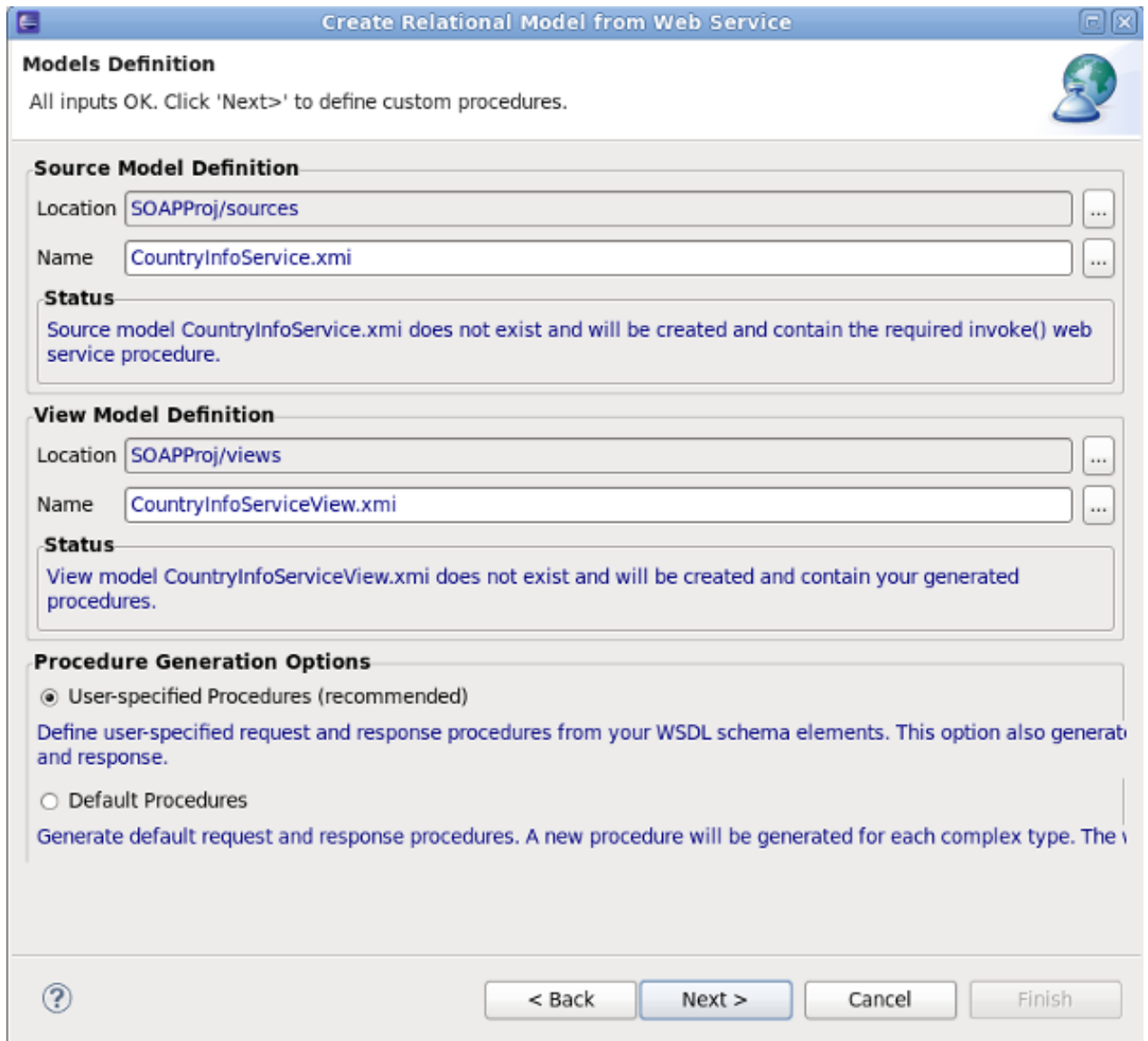


図7.25 SOAP ウィザードの消費

**Model Definition** ページでは、ソースおよびビューモデル情報のセクションが事前入力されています。ソースとビューモデルのデフォルトの名前と場所をそのまま使用します。**Next** をクリックし、以下のように次のページに進みます。

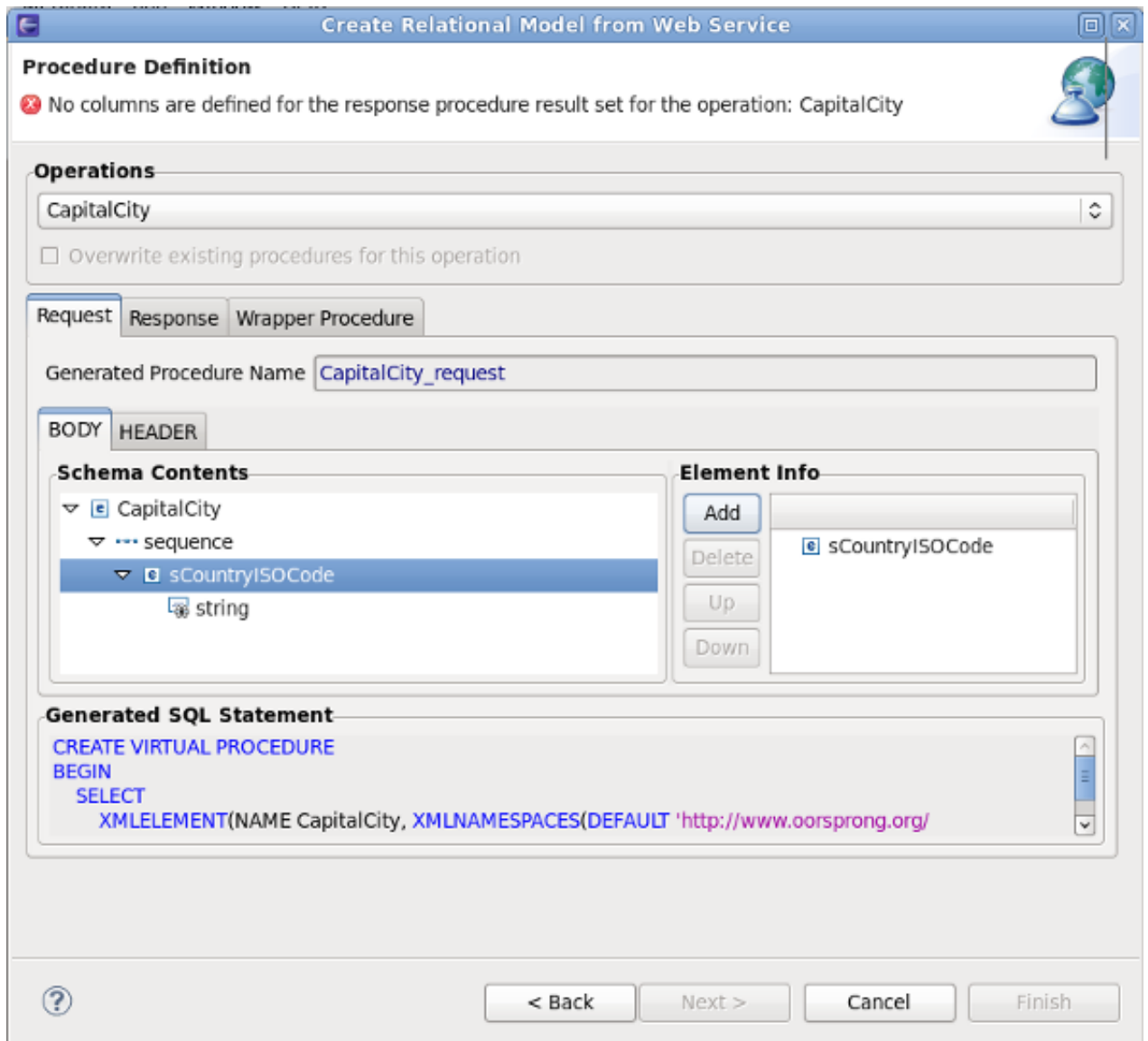


図7.26 SOAP ウィザードの消費

**Procedure Definition** ページでは、この例で唯一使用される **CapitalCity** 演算が選択されます。**Request** タブで **sCountryISOCode** 要素を選択した後、**Add** ボタンをクリックします。これで、選択された要素がリクエストに追加されます。以下のように **Response** タブを選択します。

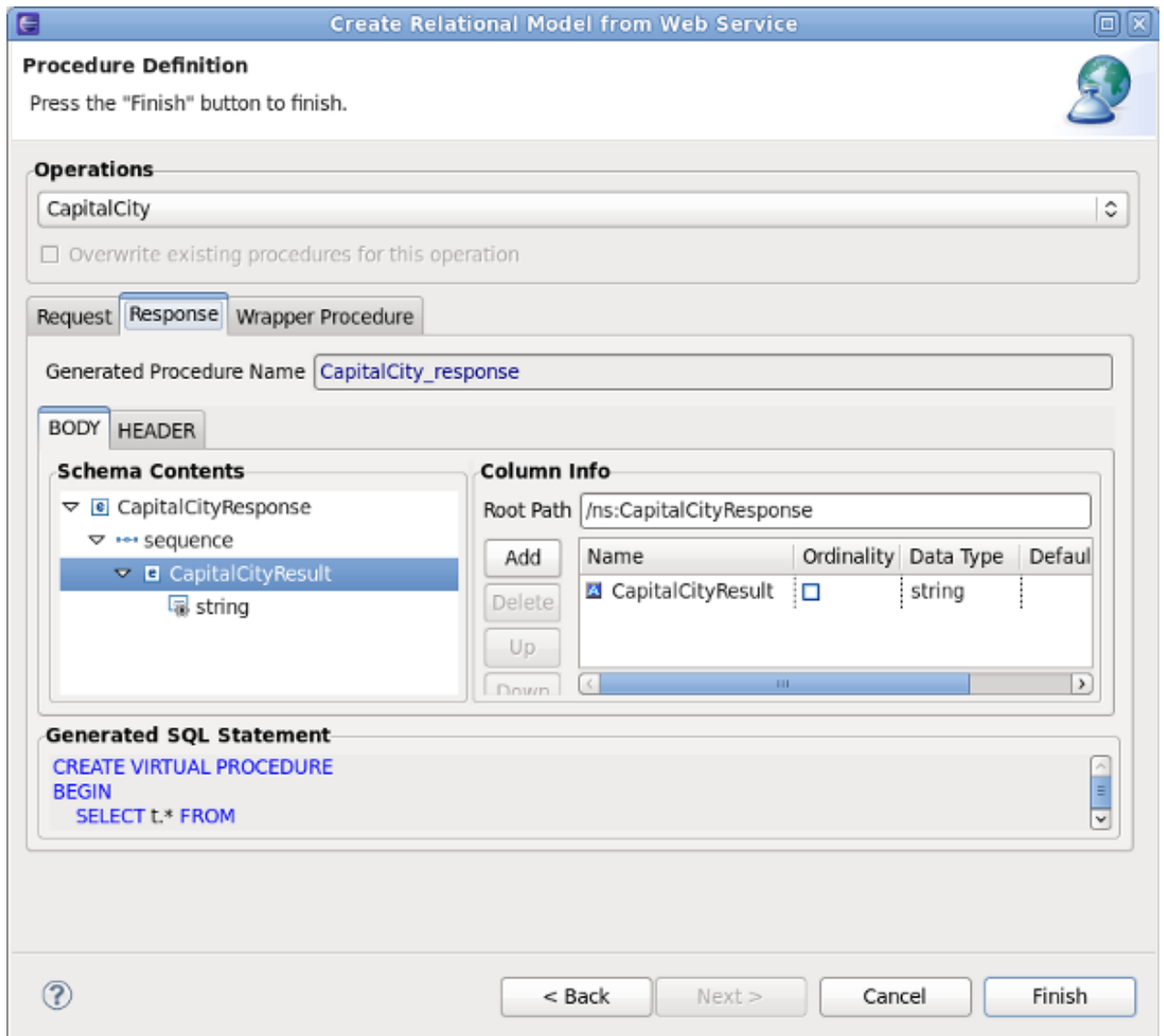


図7.27 SOAP ウィザードの消費

**Response** タブで **Body** サブタブを選択します。**Schema Contents** で **CapitalCityResult** を選択した後、**Add** ボタンをクリックします。**Wrapper Procedure** タブを選択し、以下のように完全な生成されたプロシージャの SQL を表示します。

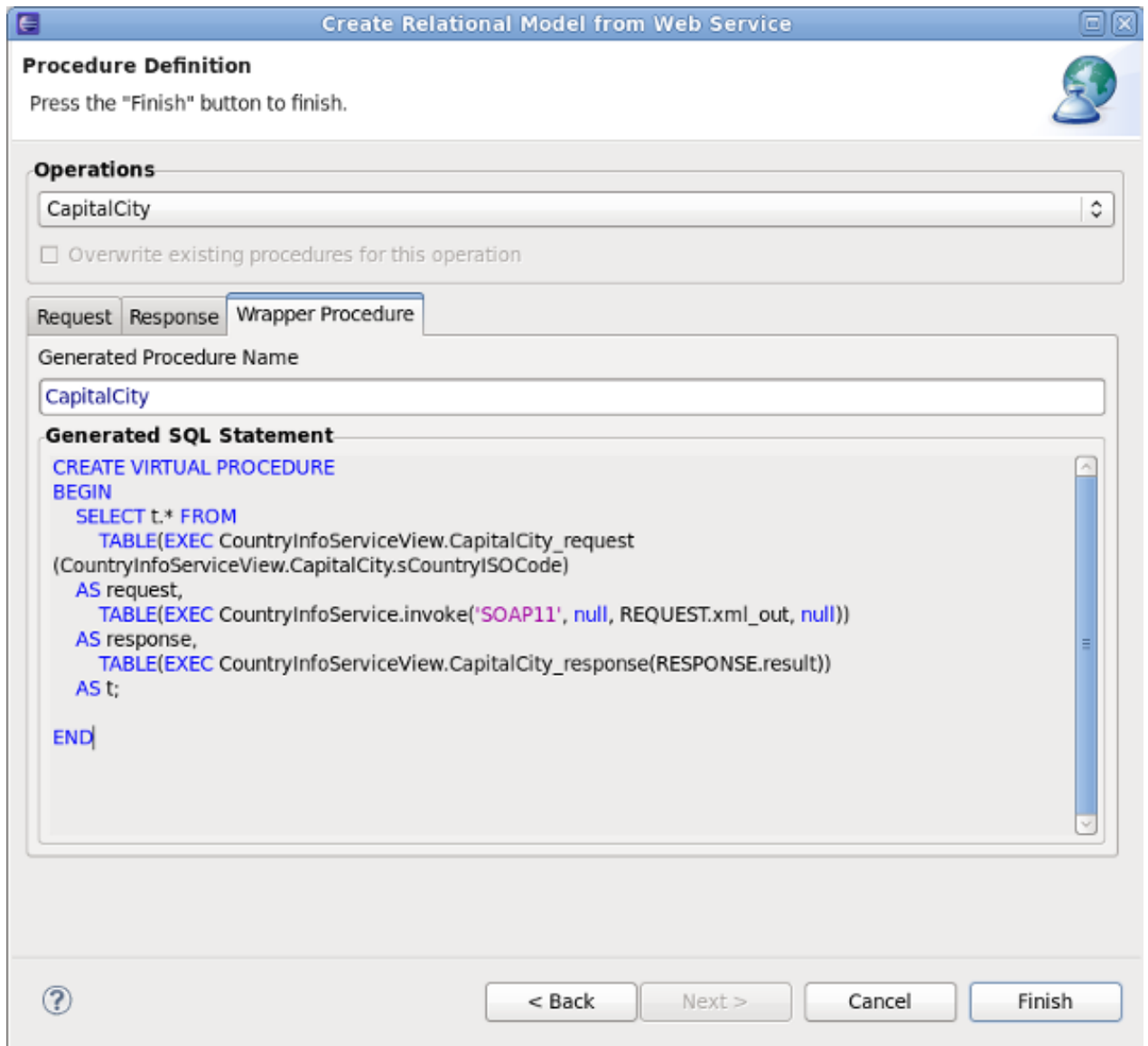




図7.28 SOAP ウィザードの消費

Click **Finish** to exit the wizard. In the Cheat Sheet, you can now continue. Click   to advance to the next step.


## 5. Create VDB


Cheat Sheet のこのセクションには、以前のステップで作成したモデルを使用して VDB を作成する手順が記載されています。Cheat Sheet のセクションは次のとおりです。






図7.29 Create VDB

Click  next to **Launch New VDB Wizard** to launch the wizard. Follow the steps to create a VDB in your workspace. When complete, exit the wizard. In the Cheat Sheet, you can now continue.

Click  to advance to the next step.

## 6. Test VDB

This final section of the Cheat Sheet provides instructions for executing the VDB created in the previous step. Click  next to **Launch Execute VDB Dialog** to launch the wizard. Select the previously created VDB to execute it.

## 第8章 New Model ウィザード

### 8.1. New Model ウィザードの起動

モデルは **Teiid Designer** によって使用される主なリソースです。モデルを作成するには、既存のメタデータを直接インポートするか、複数の New Model ウィザードのオプションの 1 つを使用します。ここでは、ウィザードについて詳しく説明します。

以下の方法の 1 つを使用して New Model ウィザードを起動します。

- ※ A. **File > New... > Metadata Model** の順にアクションをクリックします。
- B. **Model Explorer View** のプロジェクトまたはフォルダーを選択し、右クリックのメニューで同じアクションを選択します。
- C. メインツールバーの **New** ボタンを選択し、**Metadata Model** アクションを選択します。



#### 注記

**Teiid Designer** ではモデル名が一意である必要があります。New Model ウィザードおよびダイアログでモデル名を指定するとき、既存の名前を入力しないようにするためエラーメッセージが表示されます。

### 8.2. 新しいリレーショナルソースモデルの作成

#### 8.2.1. 新しいリレーショナルソースモデルの作成

新しい空のリレーショナルソースモデルを作成するには、以下を行います。

1. **New Model Wizard** を起動します。
2. 一意なモデル名を指定します。
3. **Model Class** ドロップダウンメニューから **Relational** オプションを選択します。
4. **Model Type** ドロップダウンメニューから **Source Model** を選択します。
5. **Finish** をクリックします。



#### 注記

**Browse...** ボタンを選択し、ワークスペース内でプロジェクトまたはフォルダーを選択すると、目的の場所 (例: プロジェクトまたはフォルダー) を変更できます。

新しい空のリレーショナルソースモデルを作成する他に、同じモデルクラスの既存のモデルからコピーすることもできます。

#### 8.2.2. ファイルトランスレータープロシージャの生成

このビルダーオプションを利用すると、ファイルトランスレーターを介してファイルベースのデータにアクセスするために必要な1つまたは複数のプロシージャが含まれるリレーショナルモデルを構築することができます。

ファイルトランスレータープロシージャが含まれる新しいリレーショナルモデルを作成するには、「新しいリレーショナルソースモデルの作成」に記載されている手順を行い、以下の追加ステップを実行します。

1. **Generate File Translator Procedures** というラベルが付いたモデルビルダーを選択し、**Next >** をクリックします。**Generate File Translator Procedures** ダイアログが表示されます。
2. 1つまたは複数の **Available File Translator Procedures** を選択し、**Finish** をクリックします。

### 8.2.3. Web サービストランスレータープロシージャの生成

このビルダーオプションを利用すると、web トランスレーターを介して web サービスベースの XML データにアクセスするために必要な1つまたは複数のプロシージャが含まれるリレーショナルモデルを構築することができます。

Web サービスベースのトランスレータープロシージャが含まれる新しいリレーショナルモデルを作成するには、「新しいリレーショナルソースモデルの作成」に記載されている手順を行い、以下の追加ステップを実行します。

1. **Generate Web Service Translator Procedures** というラベルが付いたモデルビルダーを選択し、**Next >** をクリックします。**Generate Web Service Translator Procedures** ダイアログが表示されます。
2. 1つまたは複数の **Available Web Services Translator Procedures** を選択し、**Finish** をクリックします。

### 8.2.4. 既存リレーショナルソースモデルからのコピー

このビルダーオプションは、新たに定義されたモデルへ既存モデルの内容の構造をコピーします。完全コピーまたは個別のモデルコンポーネントのコピーを選択できます。

別のリレーショナルソースモデルから内容をコピーして新しいリレーショナルモデルを作成するには、「新しいリレーショナルソースモデルの作成」に記載されている手順を行い、以下の追加ステップを実行します。

1. 同じモデルクラスの **Copy from existing model** というラベルが付いたモデルビルダーを選択し、**Next >** をクリックします。**Copy Existing Model** ダイアログが表示されます。
2. Browse ボタンを使用してワークスペースから既存のリレーショナルモデルを選択します。



#### 注記

New Model ウィザードを起動する前にモデルエクスプローラービューでワークスペースのリレーショナルモデルを選択した場合、既存のモデルが事前選択されます。

3. 必要に応じて **Copy all descriptions** オプションを選択します。**Finish** をクリックします。

## 8.3. 新しいリレーショナルビューモデルの作成

### 8.3.1. 新しいリレーショナルビューモデルの作成

新しい空のリレーショナルビューモデルを作成するには、以下を行います。

1. **New Model Wizard** を起動します。
2. 一意なモデル名を指定します。
3. **Model Class** ドロップダウンメニューから **Relational** オプションを選択します。
4. **Model Type** ドロップダウンメニューから **View Model** を選択します。
5. **Finish** をクリックします。



#### 注記

**Browse...** ボタンを選択し、ワークスペース内でプロジェクトまたはフォルダーを選択すると、目的の場所 (例: プロジェクトまたはフォルダー) を変更できます。

新しい空のリレーショナルビューモデルを作成する他に、以下のビルダーオプションを利用できます。

1. 同じモデルクラスの既存モデルからのコピー。
2. 既存モデルからのトランスフォーメーション。

### 8.3.2. 既存リレーショナルビューモデルからのコピー

このビルダーオプションは、新たに定義されたモデルへ既存モデルの内容の構造をコピーします。完全コピーまたは個別のモデルコンポーネントのコピーを選択できます。

別のリレーショナルビューモデルから内容をコピーして新しいリレーショナルモデルを作成するには、「新しいリレーショナルビューモデルの作成」に記載されている手順を行い、以下の追加ステップを実行します。

1. 同じモデルクラスの **Copy from existing model** というラベルが付いたモデルビルダーを選択し、**Next >** をクリックします。**Copy Existing Model** ダイアログが表示されます。
2. **Browse** ボタンを使用してワークスペースから既存のリレーショナルモデルを選択します。
3. 必要に応じて **Copy all descriptions** オプションを選択します。**Finish** をクリックします。

### 8.3.3. 既存リレーショナルビューモデルからのトランスフォーメーション

このオプションは、各ソーステーブルのデフォルトトランスフォーメーション (**SELECT \* FROM SourceModel.Table\_X**) を作成する追加機能でリレーショナルソースモデルからリレーショナルビューモデルを作成する場合のみ該当します。手順は、前述の「リレーショナルビューモデルからのコピー」と同じです。

ウィザードの2つ目のダイアログウィンドウに、リレーショナルテーブルのサポート更新プロパティを自動的に **false** に設定できる追加オプションがあります。このオプションを選択しないと、デフォルト値は **true** になります。

## 8.4. 新しい XML ドキュメントビューモデルの作成

### 8.4.1. 新しい XML ドキュメントビューモデルの作成

新しい空の XML ドキュメントビューモデルを作成するには、以下を行います。

1. **New Model Wizard** を起動します。
2. 一意なモデル名を指定します。
3. **Model Class** ドロップダウンメニューから **XML** オプションを選択します。
4. **Model Type** ドロップダウンメニューから **View Model** を選択します。
5. **Finish** をクリックします。



#### 注記

**Browse...** ボタンを選択し、ワークスペース内でプロジェクトまたはフォルダーを選択すると、目的の場所 (例: プロジェクトまたはフォルダー) を変更できます。

新しい空の XML ドキュメントビューモデルを作成する他に、以下のビルダーオプションを利用できます。

1. 同じモデルクラスの既存モデルからのコピー。
2. XML スキーマからの XML ドキュメントのビルド。

### 8.4.2. 既存 XML ドキュメントビューモデルのコピー

このビルダーオプションは、新たに定義されたモデルへ既存モデルの内容の構造をコピーします。完全コピーまたは個別のモデルコンポーネントのコピーを選択できます。

別の XML ドキュメントビューモデルから内容をコピーして新しいリレーショナルモデルを作成するには、「新しい XML ドキュメントビューモデルの作成」に記載されている手順を行い、以下の追加ステップを実行します。

1. 同じモデルクラスの **Copy from existing model** というラベルが付いたモデルビルダーを選択し、**Next >** をクリックします。**Copy Existing Model** ダイアログが表示されます。
2. **Browse** ボタンを使用してワークスペースから既存のリレーショナルモデルを選択します。
3. 必要に応じて **Copy all descriptions** オプションを選択します。**Finish** をクリックします。

### 8.4.3. XML ドキュメントビューモデルのビルド

このオプションは、選択された XML スキーマとその依存関係を基にして XML ビュードキュメントモデルを作成します。

XML スキーマから新しい XML ドキュメントビューモデルを作成するには、「XML ドキュメントビューモデルの作成」に記載されている手順を行い、以下の追加ステップを実行します。

1. **Build XML documents from XML** スキーマというラベルが付いたモデルビルダーを選択し、**Next >** をクリックします。**Select XML Schema** ダイアログが表示されます。
2. **Browse** ボタンを使用してワークスペースから既存のスキーマモデルを選択します。

## 注記

New Model ウィザードを起動する前に VDB エクスプローラーでワークスペースの XSD モデルを選択した場合、既存のモデルが事前選択されます。スキーマはワークスペースにある必要があるため、ワークスペースに 1 つまたは複数のスキーマを取り込む場合は、ファイルシステムインポーターの XSD Schemas を使用します。

3. 矢印ボタンを使用して、新しいモデルで仮想ドキュメントとしたい利用可能なスキーマのルート要素を Virtual Documents リストに移動します。
4. 適切なドキュメントオプションとマッピングオプションを選択します。 **Finish** をクリックします。
5. **Finish** をクリックして選択したすべてのドキュメントエンティティのモデルを作成します。または、オプションで **Next >** をクリックし、ドキュメントエンティティの統計を表示する **Selected Documents Statistics** ページを表示します。このページでは作成したモデルのサイズを推定できます。
6. オプションで、**Finish** をクリックして選択したすべてのドキュメントエンティティのモデルを作成するか、**Next >** をクリックし、ドキュメント固有のエンティティを除外できる **Preview Generated Documents** ページを表示します。 **Finish** をクリックします。

## 注記

深くネストされたスキーマでは、合計のエンティティ数が大きくなる可能性があります。そのため、プレビューの表示に時間がかかることがあります。

## 8.5. 新しい XML スキーマモデルの作成

### 8.5.1. 新しい XML スキーマモデルの作成

新しい空の XML スキーマ (.xsd) モデルを作成するには、以下を行います。

1. **New Model Wizard** を起動します。
2. 一意なモデル名を指定します。
3. **Model Class** ドロップダウンメニューから **XML Schema (XSD)** オプションを選択します。
4. **Model Type** ドロップダウンメニューから **Datatype Model** を選択します。
5. **Finish** をクリックします。

## 注記

**Browse...** ボタンを選択し、ワークスペース内でプロジェクトまたはフォルダーを選択すると、目的の場所 (例: プロジェクトまたはフォルダー) を変更できます。

新しい空の XML スキーマモデルを作成する他に、以下のビルダーオプションを利用できます。

- ※ 同じモデルクラスの既存モデルからのコピー。

## 8.5.2. 既存 XML スキーマモデルのコピー

このビルダーオプションは、新たに定義されたモデルへ既存モデルの内容の構造をコピーします。完全コピーまたは個別のモデルコンポーネントのコピーを選択できます。

別の XML スキーマモデルから内容をコピーして新しいリレーショナルモデルを作成するには、「新しい XML スキーマモデルの作成」に記載されている手順を行い、以下の追加ステップを実行します。

1. 同じモデルクラスの **Copy from existing model** というラベルが付いたモデルビルダーを選択し、**Next >** をクリックします。**Copy Existing Model** ダイアログが表示されます。
2. Browse ボタンを使用してワークスペースから既存のリレーショナルモデルを選択します。
3. 必要に応じて **Copy all descriptions** オプションを選択します。**Finish** をクリックします。

## 8.6. 新しい Web サービスビューモデルの作成

### 8.6.1. 新しい Web サービスビューモデルの作成

新しい空の Web サービスビューモデルを作成するには、以下を行います。

1. **New Model Wizard** を起動します。
2. 一意なモデル名を指定します。
3. **Model Class** ドロップダウンメニューから **Web Service** オプションを選択します。
4. **Model Type** ドロップダウンメニューから **View Model** を選択します。
5. **Finish** をクリックします。



#### 注記

**Browse...** ボタンを選択し、ワークスペース内でプロジェクトまたはフォルダーを選択すると、目的の場所 (例: プロジェクトまたはフォルダー) を変更できます。

新しい空の Web サービスビューモデルを作成する他に、以下のビルダーオプションを利用できます。

1. 同じモデルクラスの既存モデルからのコピー。
2. 既存の WSDL ファイルまたは URL からのビルド。

### 8.6.2. 既存の Web サービスビューモデルのコピー

このビルダーオプションは、新たに定義されたモデルへ既存モデルの内容の構造をコピーします。完全コピーまたは個別のモデルコンポーネントのコピーを選択できます。

別の Web サービスビューモデルから内容をコピーして新しいリレーショナルモデルを作成するには、「新しい Web サービスビューモデルの作成」に記載されている手順を行い、以下の追加ステップを実行します。

1. 同じモデルクラスの **Copy from existing model** というラベルが付いたモデルビルダーを選択し、**Next >** をクリックします。**Copy Existing Model** ダイアログが表示されます。
2. Browse ボタンを使用してワークスペースから既存のリレーショナルモデルを選択します。
3. 必要に応じて **Copy all descriptions** オプションを選択します。**Finish** をクリックします。

### 8.6.3. WSDL ファイルまたは URL からの Web サービスビューモデルの作成

このビルダーオプションは、ユーザー定義の WSDL ファイルとその参照されたスキーマを基にして Web サービスモデルを作成します。また、該当する XML スキーマファイルおよび XML ビュードキュメントモデル (任意) も作成されます。

別の Web サービスビューモデルから内容をコピーして新しいリレーショナルモデルを作成するには、「新しい Web サービスビューモデルの作成」に記載されている手順を行い、以下の追加ステップを実行します。

1. **Build from existing WSDL file(s) or URL** というラベルの付いたモデルビルダーを選択し、**Next >** をクリックします。
2. ウィザードの残りの手順は、「WSDL の Web サービスへのインポート」に記載されているオプションの手順と同じです。

### 8.6.4. XML ドキュメントビューモデルからの Web サービスビューモデルの作成

**Teiid Designer** では、Web サービスモデルとそれらに対応するインターフェースおよび演算を XML ビューモデルコンポーネント (XML ビュードキュメントおよび XML ビュードキュメントルート) から生成できます。

XML コンポーネントから新しい Web サービスモデルを作成するには、以下を行います。

1. Model Explorer ビューで、単一の XML ドキュメントまたは単一の XML ドキュメントルートを選択します。
2. 右クリックして **Modeling > Create Web Service** アクションを選択します。
3. **Web Service Generation Wizard** で、不足しているプロパティを入力します。
4. **Finish** をクリックしてモデルを生成します。モデルの生成が完了すると確認ダイアログが表示されます。**OK** をクリックします。



#### 注記

Web サービスモデルおよびインターフェース名の値を変更し (「...」 ボタンを使用)、既存の Web サービスモデルコンポーネントを使用できます。これにより、既存モデルに新しい演算が作成されます。



## 第9章 インポーター

### 9.1. インポーター

Import ウィザードは、データソースの構造を基にモデルを作成する方法を提供し、既存のメタデータ (WSDL または XML スキーマ) をソースモデルへ変換したり、既存のメタデータファイルを現在の VDB へロードしたりします。

Import ウィザードを起動するには、**File > Import** アクションを選択するかツリーのプロジェクト、フォルダー、またはモデルを選択し、右クリックで **Import...** を選択します。

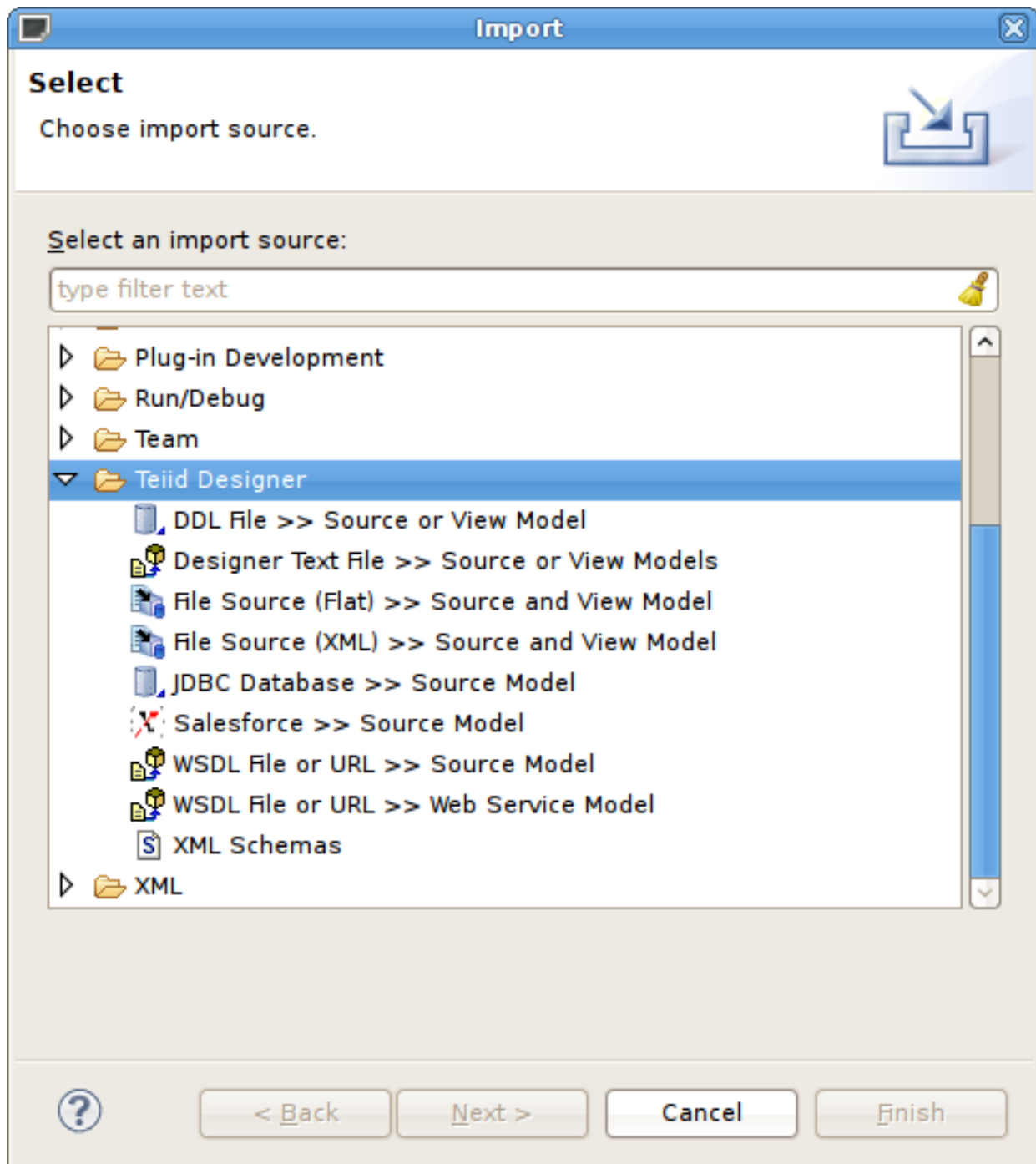
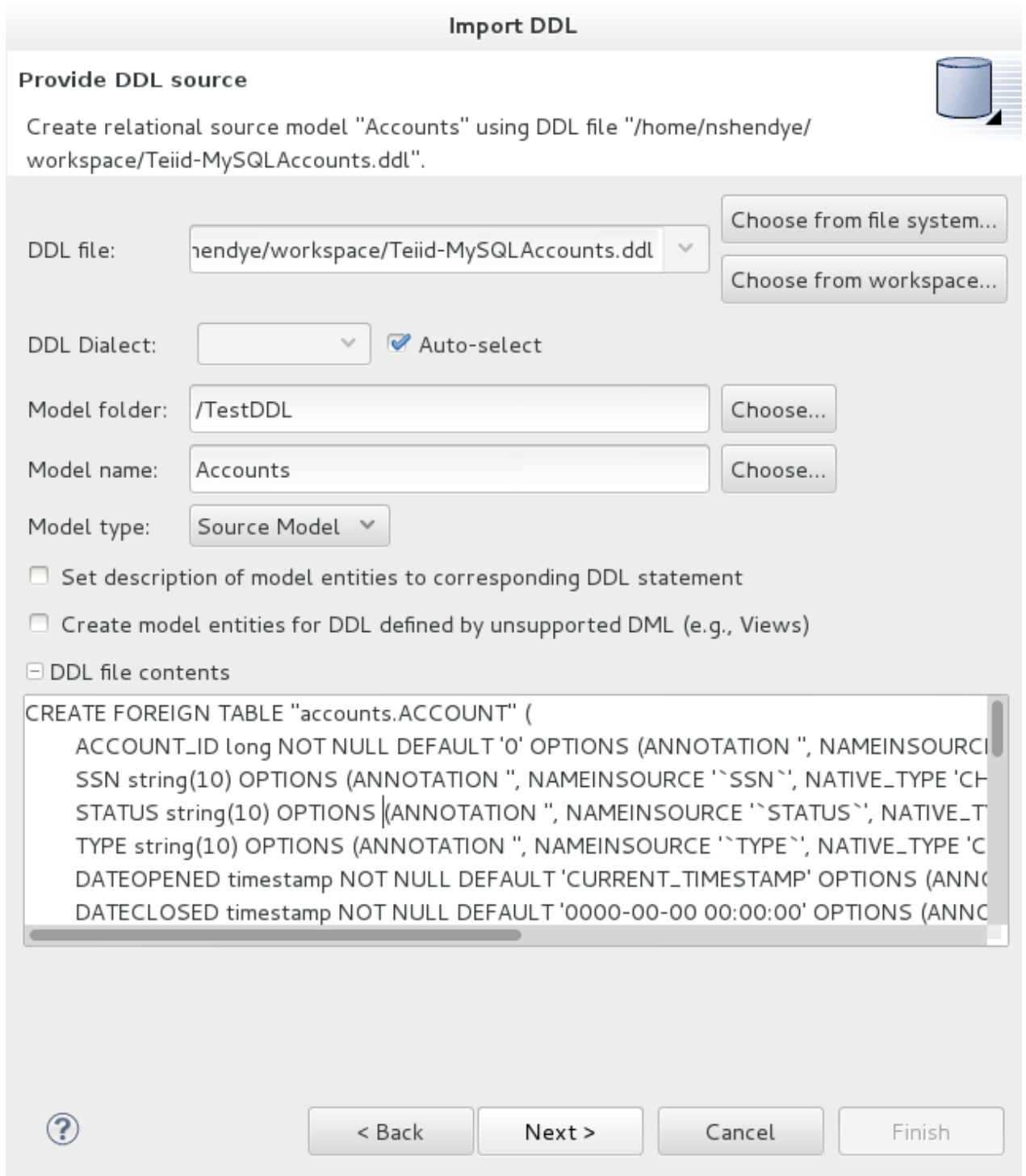


図9.1 Import ウィザード

### 9.2. DDL のインポート

以下の手順に従って DDL をインポートすると、ソースリレーショナルモデルを作成できます。

1. In Model Explorer, click **File** > **Import** action  in the toolbar or select a project, folder or model in the tree and click **Import...**
2. インポートオプション **Teiid Designer** > **DDL File** >> **Source or View Model** を選択し、**Next>** をクリックします。
3. **Choose from file system...** または **Choose from workspace...** から既存の DDL を選択します。モデルフォルダーの場所を設定し、有効なモデル名を入力または選択します。モデルタイプ (Source Model または View Model) を設定し、必要なオプションを設定してから **NEXT>** (または有効な場合は **Finish**) をクリックします。



**Import DDL**

**Provide DDL source**

Create relational source model "Accounts" using DDL file "/home/nshendye/workspace/Teiid-MySQLAccounts.ddl".

DDL file:

DDL Dialect:   Auto-select

Model folder:

Model name:

Model type:

Set description of model entities to corresponding DDL statement

Create model entities for DDL defined by unsupported DML (e.g., Views)

DDL file contents

```
CREATE FOREIGN TABLE "accounts.ACCOUNT" (
  ACCOUNT_ID long NOT NULL DEFAULT '0' OPTIONS (ANNOTATION "", NAMEINSOURCE "ACCOUNT_ID", NATIVE_TYPE 'LONG'),
  SSN string(10) OPTIONS (ANNOTATION "", NAMEINSOURCE "`SSN`", NATIVE_TYPE 'CHAR'),
  STATUS string(10) OPTIONS (ANNOTATION "", NAMEINSOURCE "`STATUS`", NATIVE_TYPE 'CHAR'),
  TYPE string(10) OPTIONS (ANNOTATION "", NAMEINSOURCE "`TYPE`", NATIVE_TYPE 'CHAR'),
  DATEOPENED timestamp NOT NULL DEFAULT 'CURRENT_TIMESTAMP' OPTIONS (ANNOTATION "", NAMEINSOURCE "DATEOPENED", NATIVE_TYPE 'TIMESTAMP'),
  DATECLOSED timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' OPTIONS (ANNOTATION "", NAMEINSOURCE "DATECLOSED", NATIVE_TYPE 'TIMESTAMP')
```

図9.2 DDL インポートオプション

4. **NEXT>** をクリックすると、個別のリレーショナルエンティティを確認または選択解除するために相違点のレポートが表示されます。**Finish** をクリックして終了します。

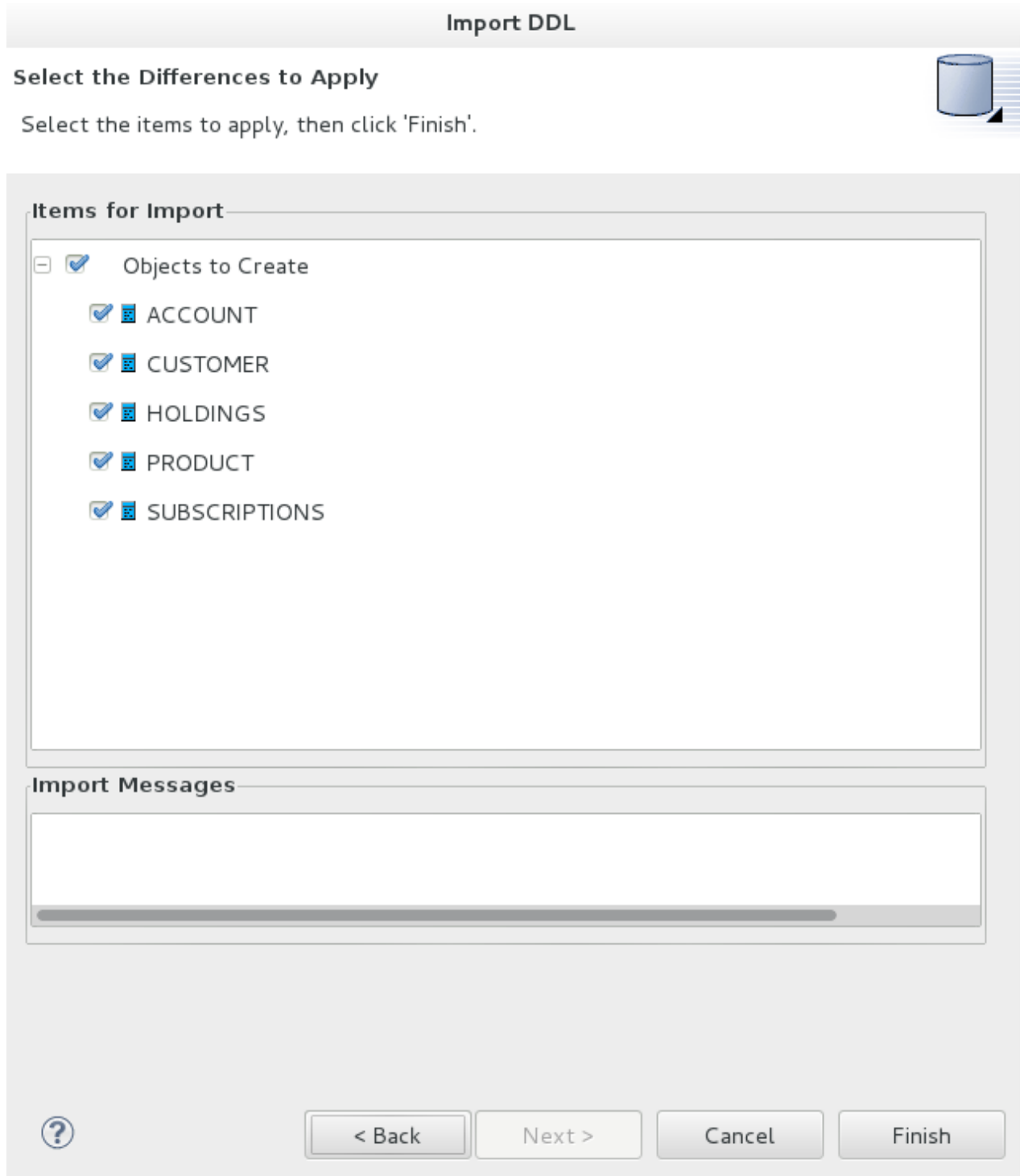


図9.3 DDL ダイアログのインポート

### 9.3. JDBC データベースからのインポート

JDBC ソーススキーマデータからリレーショナルソースモデルを作成するには、以下の手順に従います。


**注記**

データベース接続 URL 情報およびスキーマで提供される詳細によっては、ステップ 5 から 7 までを行う必要がない場合があります。

1. Model Explorer でツールバーの **File > Import** アクションをクリックするか、ツリーのプロジェクト、フォルダー、またはモデルを選択し、**Import...** をクリックします。
2. インポートオプション **Metadata Modeling > JDBC Database >> Source Model** を選択し、**Next>** をクリックします。
3. ドロップダウンメニューから既存または以前の接続プロファイルを選択するか、**New...** ボタンをクリックして **New Connection Profile** ダイアログを開始します。または、**Edit...** をクリックして選択前に既存の接続プロファイルを変更します。


**注記**

接続プロファイル選択リストは JDBC データベース接続のみで入力されます。

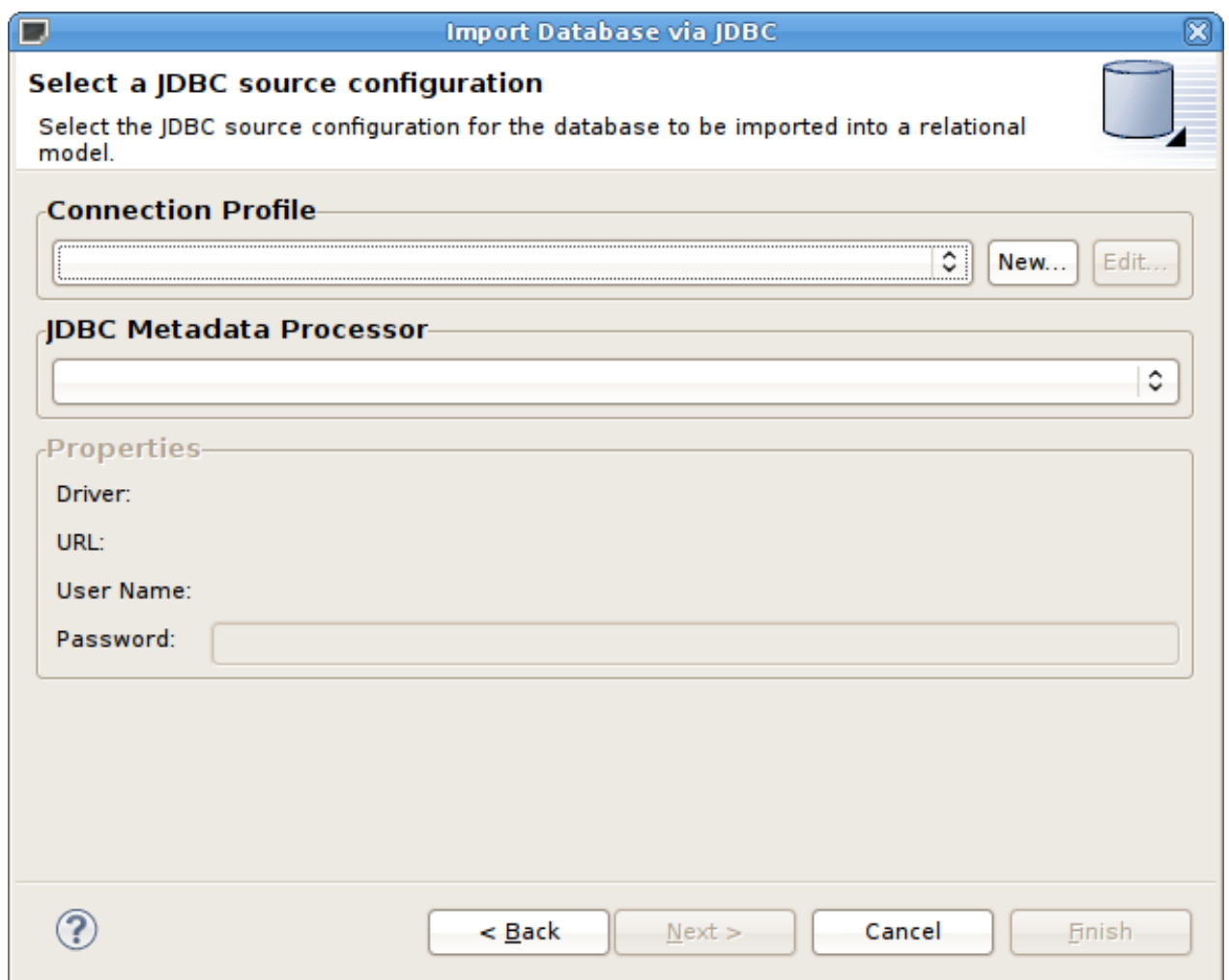


図9.4 Select JDBC Source Configuration ダイアログ

JDBC データベースは異なるため、データタイプの変換またはメタデータの解釈を行うためにメタデータの特別な処理が必要になります。JDBC メタデータプロセッサドロップダウンメニューは、選択された接続プロファイルを基に自動的に選択されます。DB2、ModeShape、ODBC、Oracle、PostgreSQL、SQL Server、および Sybase 用の特別なプロセッサを利用できます。他のデータベースではデフォルトの JDBC プロセッサを使用できます。

4. 接続プロファイルを選択した後、パスワードを入力します (入力済みでない場合)。**Next>** (または有効な場合は **Finish**) をクリックします。

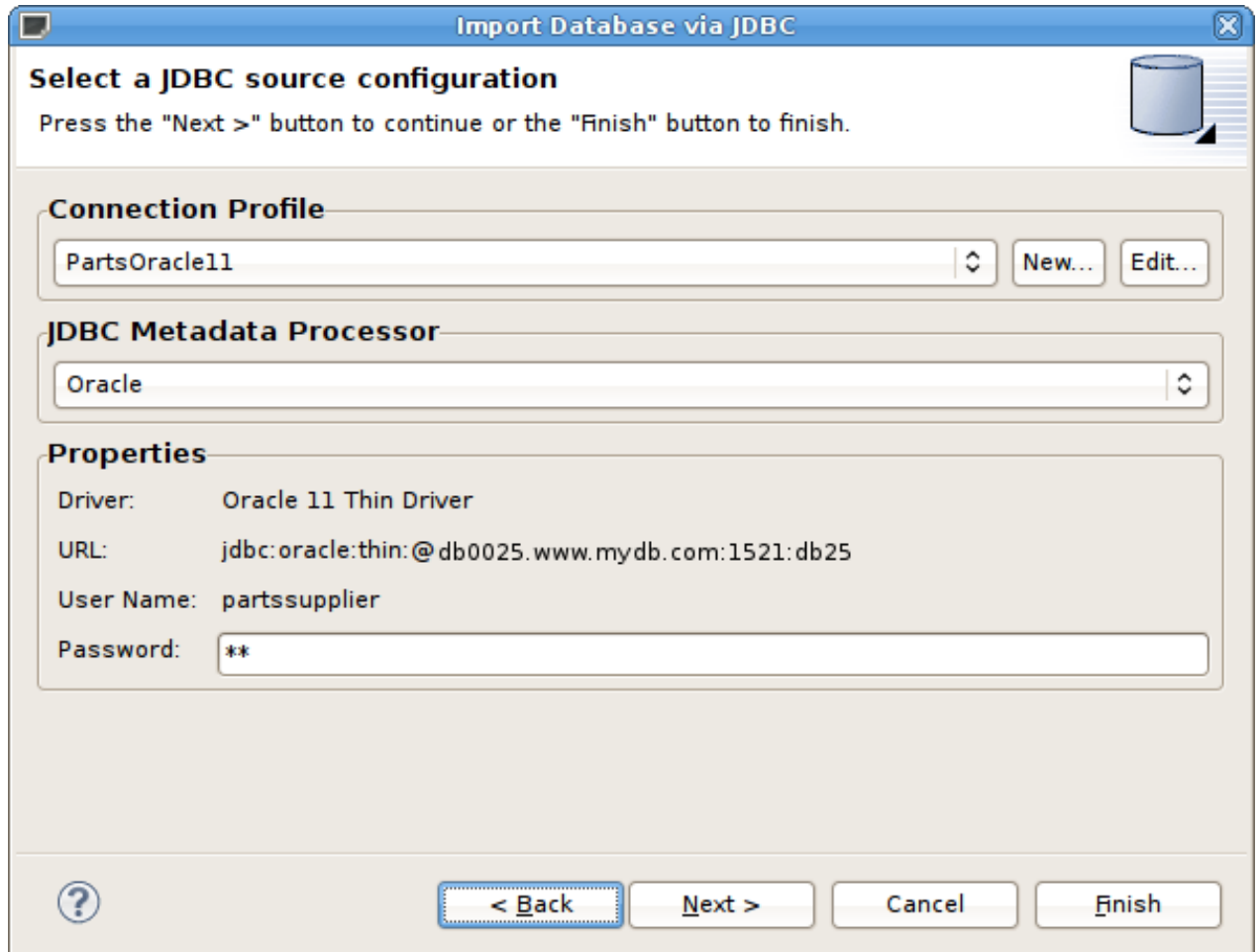


図9.5 Select JDBC Source Configuration ダイアログ

5. **Select Database Metadata** ページで、インポートするデータベースのオブジェクトのタイプを選択します。**Next>** (または有効な場合は **Finish**) をクリックします。

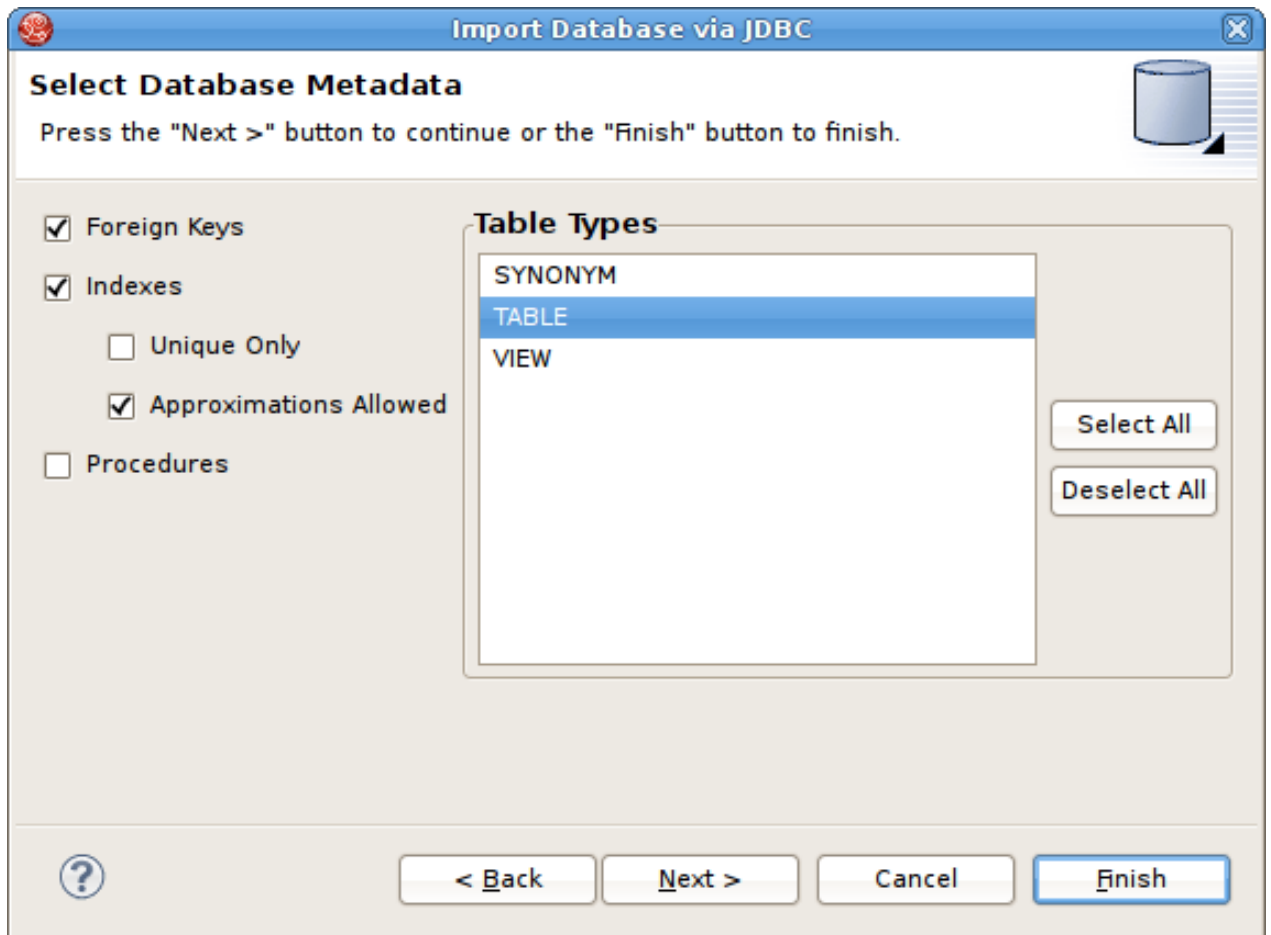


図9.6 Select Database Metadata ダイアログ

6. **Select Database Objects** ページで、スキーマの内容の表示または選択の変更を行います。リレーショナルオブジェクトの構築に使用されるデータベーススキーマオブジェクトを選択します。Next> (または有効な場合は Finish) をクリックします。

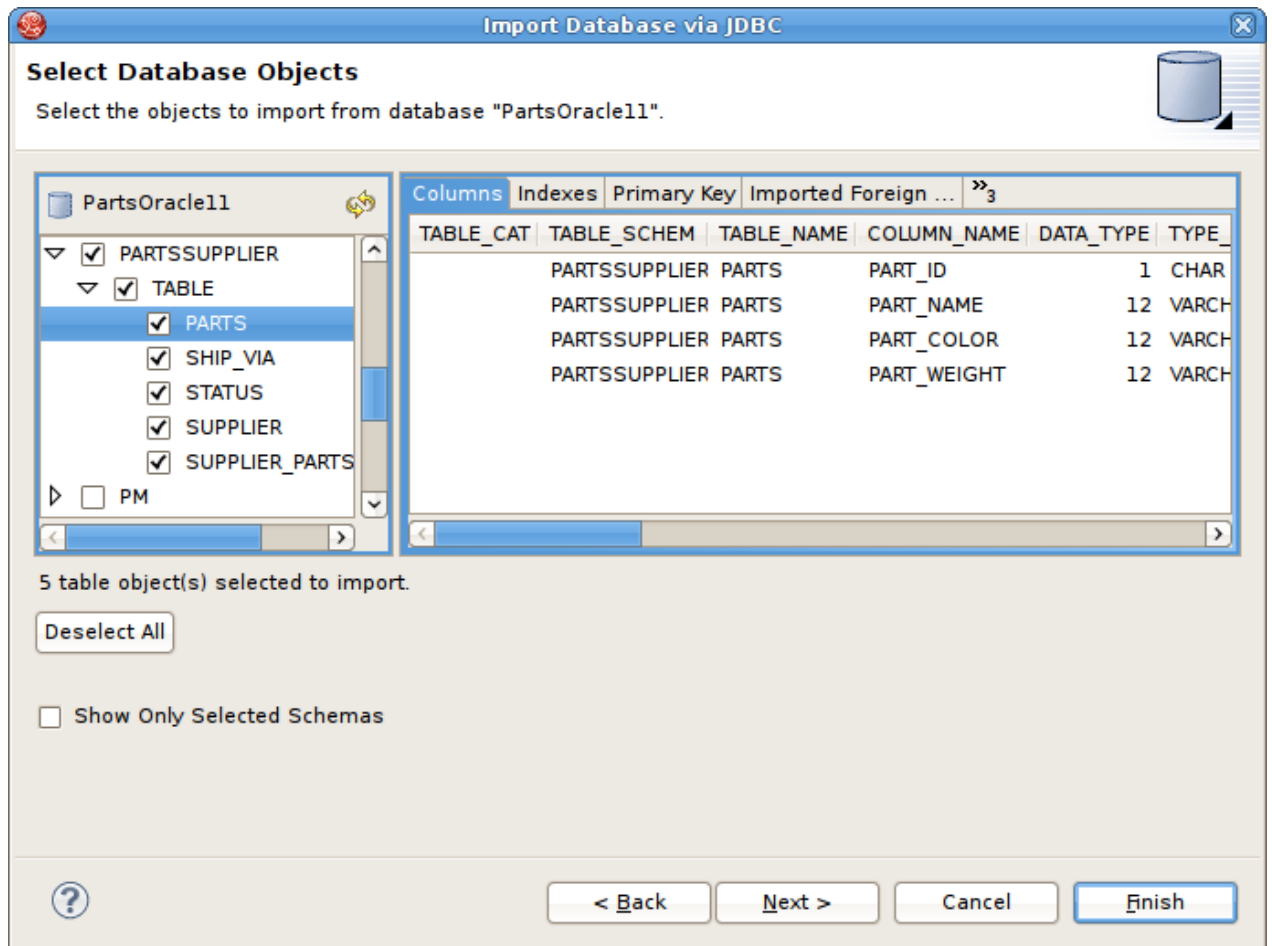


図9.7 Select Database Options ダイアログ

7. **Specify Import Options** ページで、希望のモデル名と、リレーショナルオブジェクトの名前をカスタマイズするために使用されるその他のオプションを指定します。**Finish** をクリックして終了します。

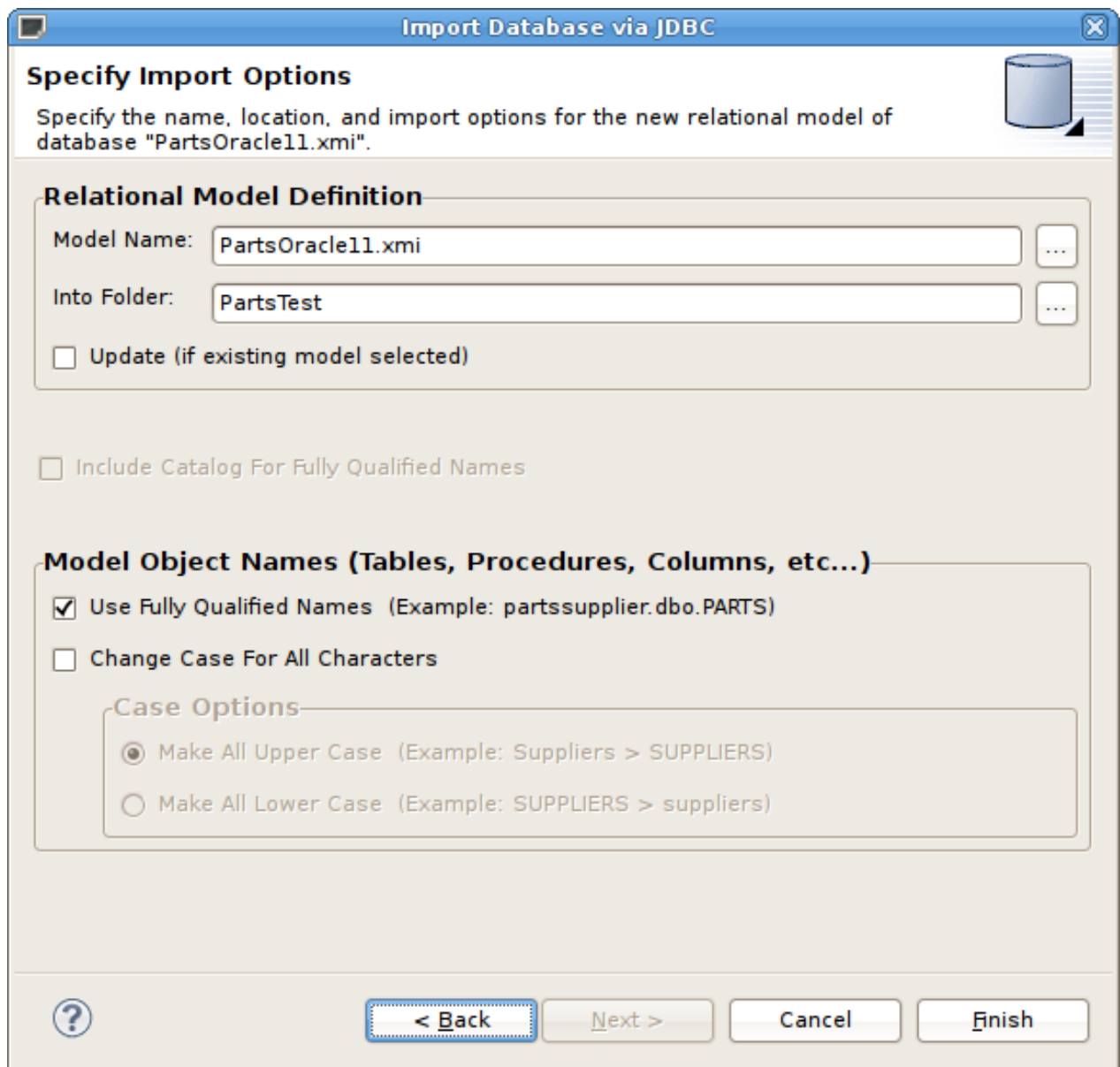


図9.8 Specify Import Options ダイアログ

8. 終了処理中に、モニターが表示され、インポートの進捗に関するフィードバックが提供されます。

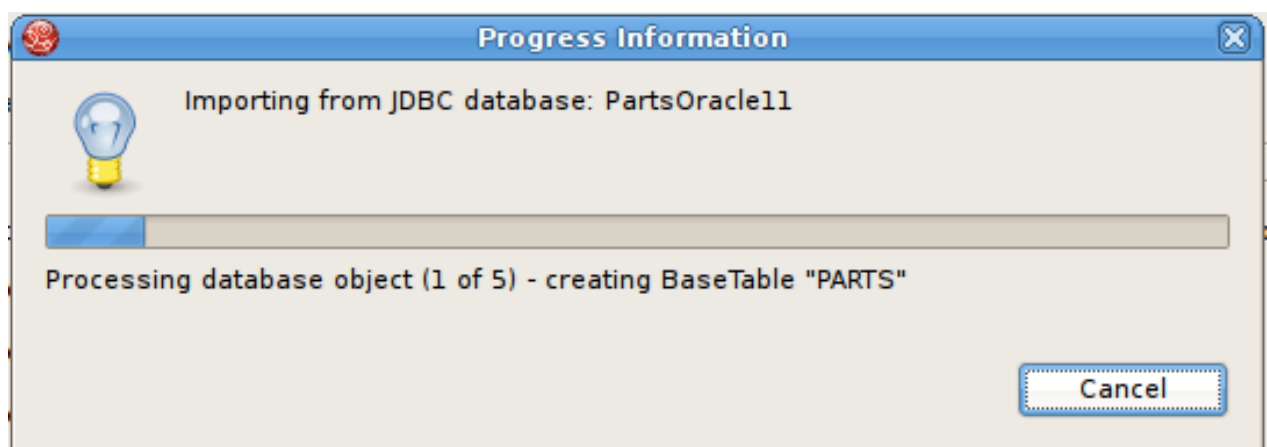


図9.9 JDBC Import Progress ダイアログ

## 9.4. Teiid データソース接続からのインポート



Teiid 接続ソースモデルのインポートオプションは、JDBC および 他の Teiid Designer インポーターでサポートされない他のデプロイされたデータソースからリレーショナルソースモデルを作成する方法を提供します。

### 手順9.1 Teiid データソース接続からのインポート

1. **Model Explorer** に移動します。
2. ツールバーの **File - Import** アクションを選択します。  
ツリーでプロジェクト、フォルダー、またはモデルを選択した後、**Import...** を選択します。
3. **Teiid Designer - Teiid Connection - Source Model** と選択し、**Next** をクリックします。
4. インポートに使用するデータソースを選択します。  
必要なソースが存在しない場合は新しいソースを作成できます。また、他のソース管理機能を実行することもできます。
5. **Next** をクリックします。
6. データソースタイプの適切なトランスレーターを選択し、作成または更新したいターゲットリレーショナルモデルを選択します。**Next** をクリックします。
7. ウィザードの次のページに移動したときに、一時的な動的 VDB が実際にサーバーにデプロイされ、データソースのスキーマが DDL で読み出されます。この DDL は表示されます (必要な場合はエクスポートすることもできます)。**Next** をクリックします。
8. ウィザードの最後のページで、個別のリレーショナルエンティティーを確認または選択解除するために相違点のレポートが表示されます。**Finish** をクリックして終了します。

## 9.5. フラットファイルソースからのインポート

フラットファイルデータソースからメタデータをインポートし、データを数分でクエリーするために必要なメタモデルを作成することができます。以下の手順に従って、フラットファイルデータソースの定義、フラットファイルの解析パラメーターの設定、標準的な Teiid フラットファイルプロジェクターが含まれるソースモデルの作成、およびフラットファイルの列データを定義する SQL が含まれるビューテーブルの作成を行います。

**JBoss Data Virtualization** はフラットファイルをデータソースとしてサポートします。**Teiid Designer** は、フラットファイルのデータにアクセスするために必要なメタデータモデルの作成を容易にする Import ウィザードを提供します。Teiid Designer の JDBC、Salesforce、および WSDL インポーターと同様に、フラットファイルインポーターは特定のデータツール接続プロファイルの使用を基にします。

インポーターの結果には、**JBoss Data Virtualization** によってサポートされる **getTextFiles()** プロシージャが含まれるソースモデルが含まれます。

インポーターは、選択したフラットファイルソースファイルのビューテーブルが含まれる新しいビューモデルも作成します。ソースモデルからの **getTextFiles()** プロシージャが含まれる SQL トランスフォーメーションはビューテーブル内で生成され、Teiid **TEXTTABLE()** 関数に必要な列定義およびパラメーターはデータファイルのクエリーに使用されます。また、新しいビューモデルを作成する代わりに既存のビューモデルの更新を選択することもできます。

**TEXTTABLE** 関数は文字入力をプロセスし、表形式出力を作成します。この関数は、固定および区切りのファイル形式解析をサポートします。関数自体は投影する列を定義します。**TEXTTABLE** 関数は暗黙的にはネストされたテーブルで、先行の FROM 句エンタリーに関連付けることがあります。

```
TEXTTABLE(expression COLUMNS <COLUMN>, ... [DELIMITER char] [(QUOTE|ESCAPE) char] [HEADER [integer]] [SKIP integer]) AS name
```

**Teiid Designer** は各ビューテーブルの完全な SQL ステートメントを次の形式で構築します。

```
SELECT A.Name, A.Sport, A.Position, A.Team, A.City, A.StateCode,
A.AnnualSalary FROM (EXEC PlayerDataSource.getTextFiles('PlayerData.txt'))
AS f, TEXTTABLE(f.file COLUMNS Name string, Sport string, Position string,
Team string, City string, StateCode string, AnnualSalary string HEADER 2
SKIP 3) AS A
```

フラットファイルソースからインポートするには、以下の手順に従います。

1. In **Model Explorer**, click **File > Import** action  in the toolbar or select a project, folder or model in the tree and click **Import...**
2. インポートオプション **Teiid Designer > File Source (Flat) >> Source and View Model** を選択し、**Next>** をクリックします。

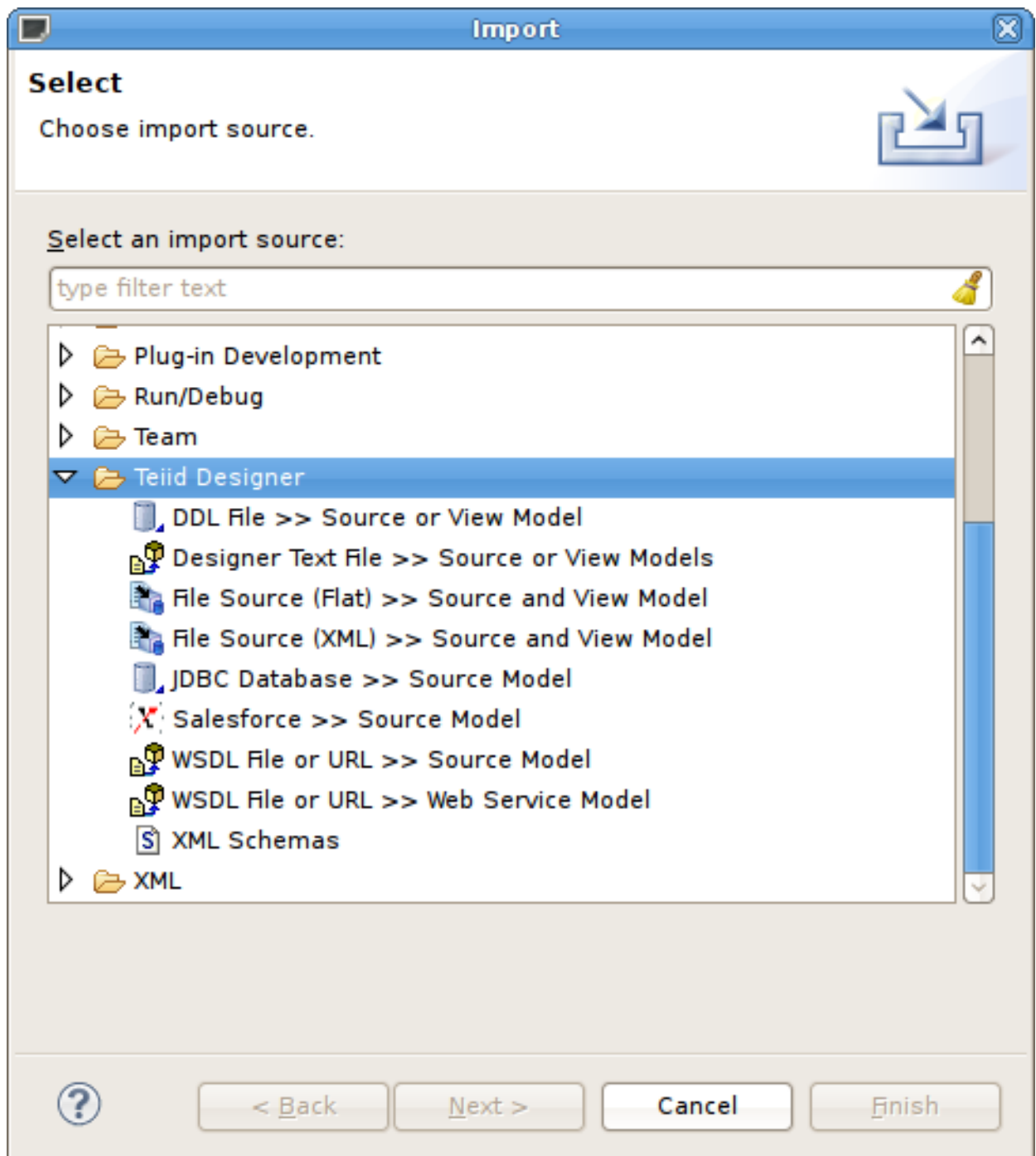


図9.10 フラットファイルソースからのインポート

3. ドロップダウンメニューから既存または以前の接続プロファイルを選択するか、**New...** ボタンをクリックして **New Connection Profile** ダイアログを開始します。または、**Edit...** をクリックして選択前に既存の接続プロファイルを変更します。Flat File Source 選択リストは、フラットファイル接続プロファイルでのみ入力されます。

接続プロファイルを選択した後、接続プロファイルで定義されたフォルダーのファイルが **Available Data Files** パネルに表示されます。処理するデータファイルを選択します。データの読み出しと結果セットの返信に必要な SQL トランスフォーメーションが含まれるビューテーブルの構築には、このファイルからのデータとカスタムインポートオプションが使用されます。

最後に、ページの下部にある **Source Model Definition** に一意なソースモデル名を入力するか、**Browse** ボタンを使用して既存のソースモデルを選択します。モデル名の有効性、モデルが存在するかどうか、およびモデルに `getTextFiles()` プロシージャがすでに含まれているかどうかを示す Model Status を確認してください。この場合、ソースモデルもプロシージャも生成され

ません。

このページの作業を終了したら **Next>** をクリックします。

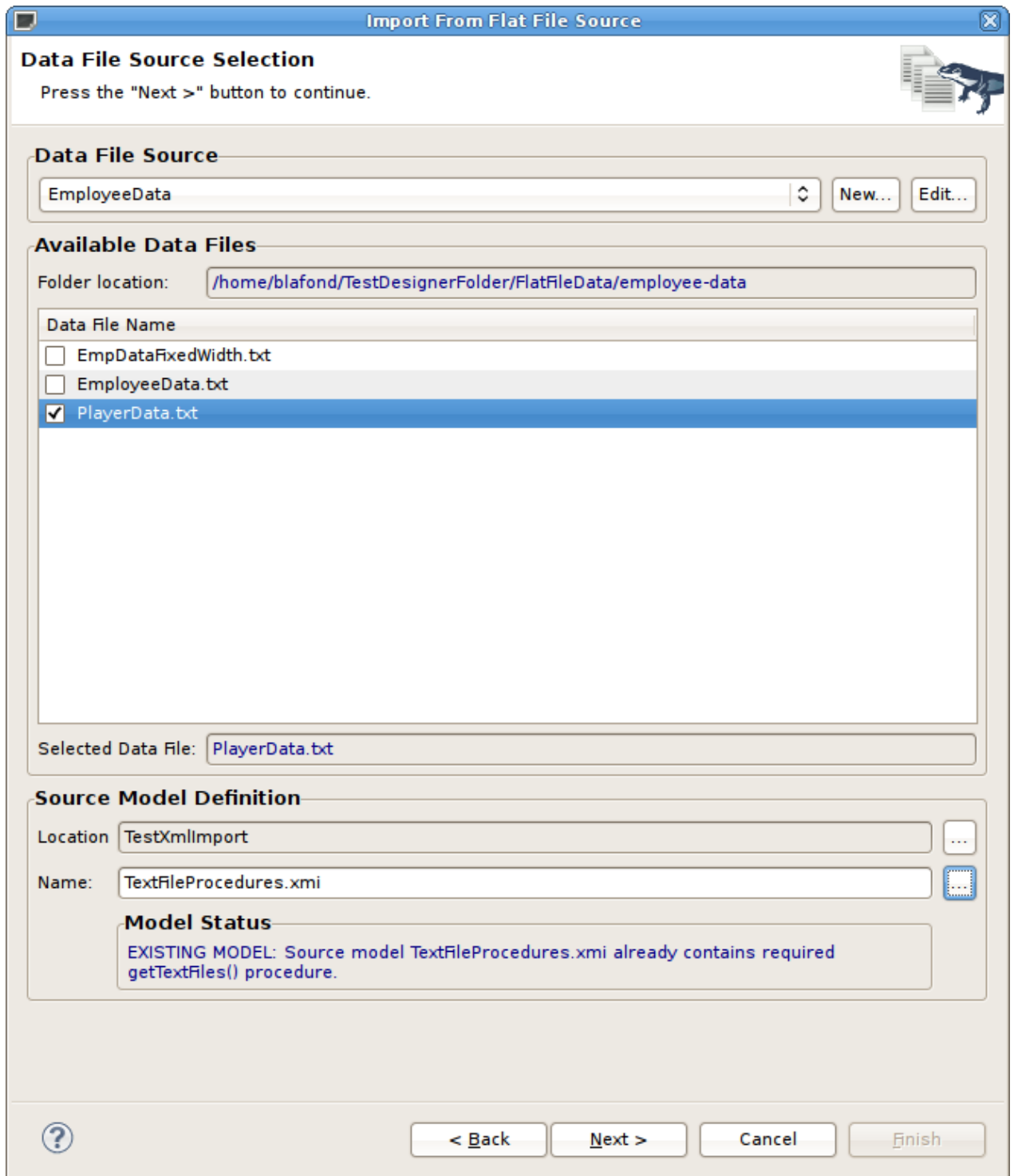


図9.11 Data File Source Selection ページ

- Flat File Column Format Definition というタイトルの次のページでは、ファイルの列データの形式を定義する必要があります。オプションは **Character delimited** と **Fixed width** です。このページにはファイル内容のプレビューが含まれ、形式を決定するのに役立ちます。最初の 20 行を表示するのがウィザードのデフォルトになりますが、必要に応じて変更できます。

このページの作業を終了したら **Next>** をクリックします。

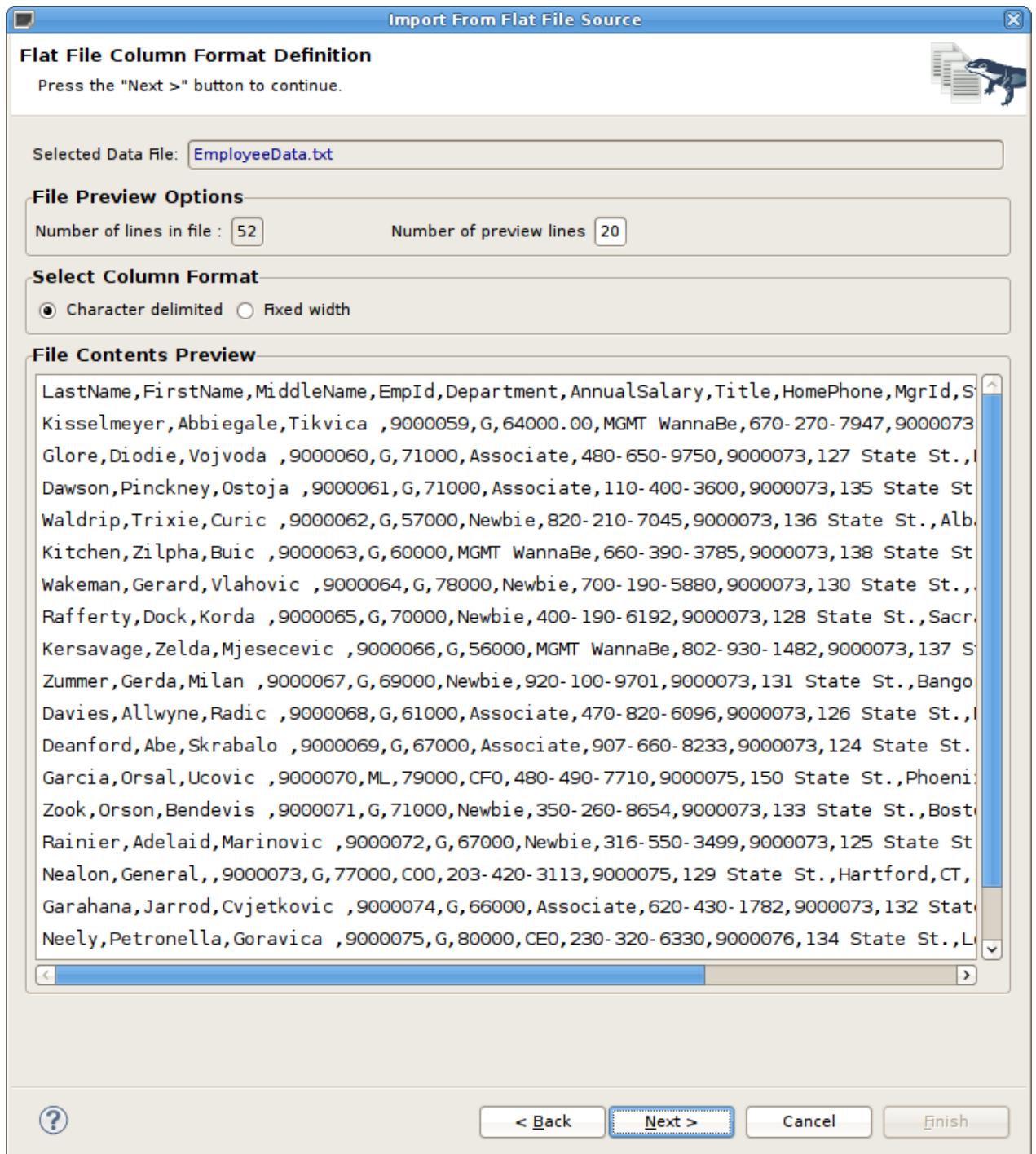


図9.12 Data File Source Selection ページ

5. Character Delimited (文字区切り) オプション - このインポーターの主な目的は、ユーザー定義のデータファイルをクエリーするために必要なトランスフォーメーションが含まれるビューテーブルを作成することです。このページには、Character delimited オプションの下のパネル (Generated SQL Statement) に表示される生成された SQL ステートメントをカスタマイズするために使用できるオプションが示されます。ヘッダーオプション (ヘッダーの列名、ヘッダー行番号、および最初のデータ行番号) の指定、選択した行の解析、文字デリミターの変更、および **TEXTTABLE()** 関数オプションの編集を行います。

ファイルヘッダーで列名が定義されていない場合や、カスタム列を編集または作成したい場合は、**ADD**、**DELETE**、**UP**、または **DOWN** を使用して SQL の列情報を管理できます。

このページの作業を終了したら **Next>** をクリックします。

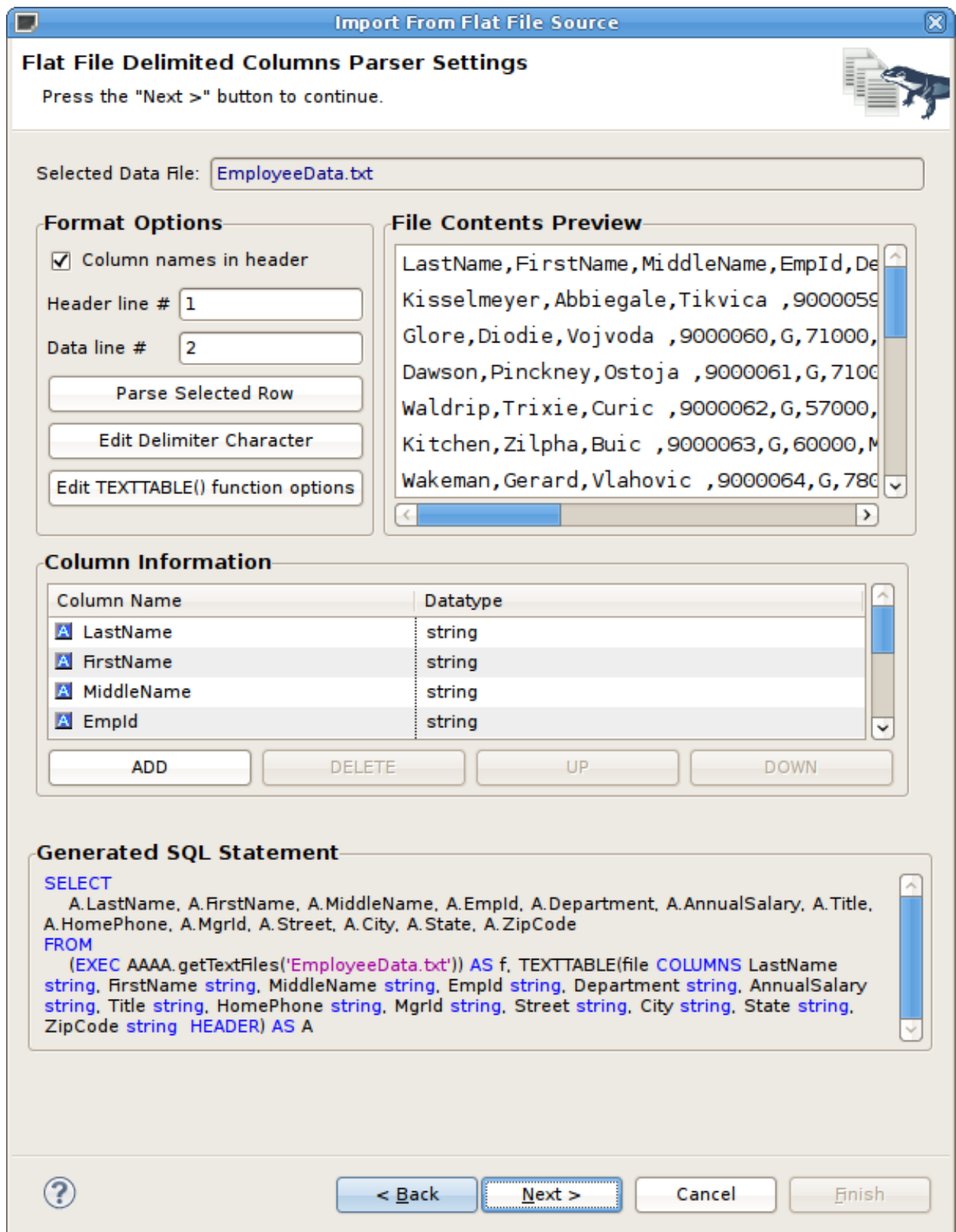


図9.13 Flat File Delimited Columns Options ページ

パーサー設定が適切であることを判断できるようにするには、**File Contents Preview** でデータ行を選択し、**Parse Selected Row** ボタンをクリックします。列のリストと結果となる列データを示すダイアログが表示されます。列データが予想どおりでなかった場合、結果に従って設定を調整する必要があります。

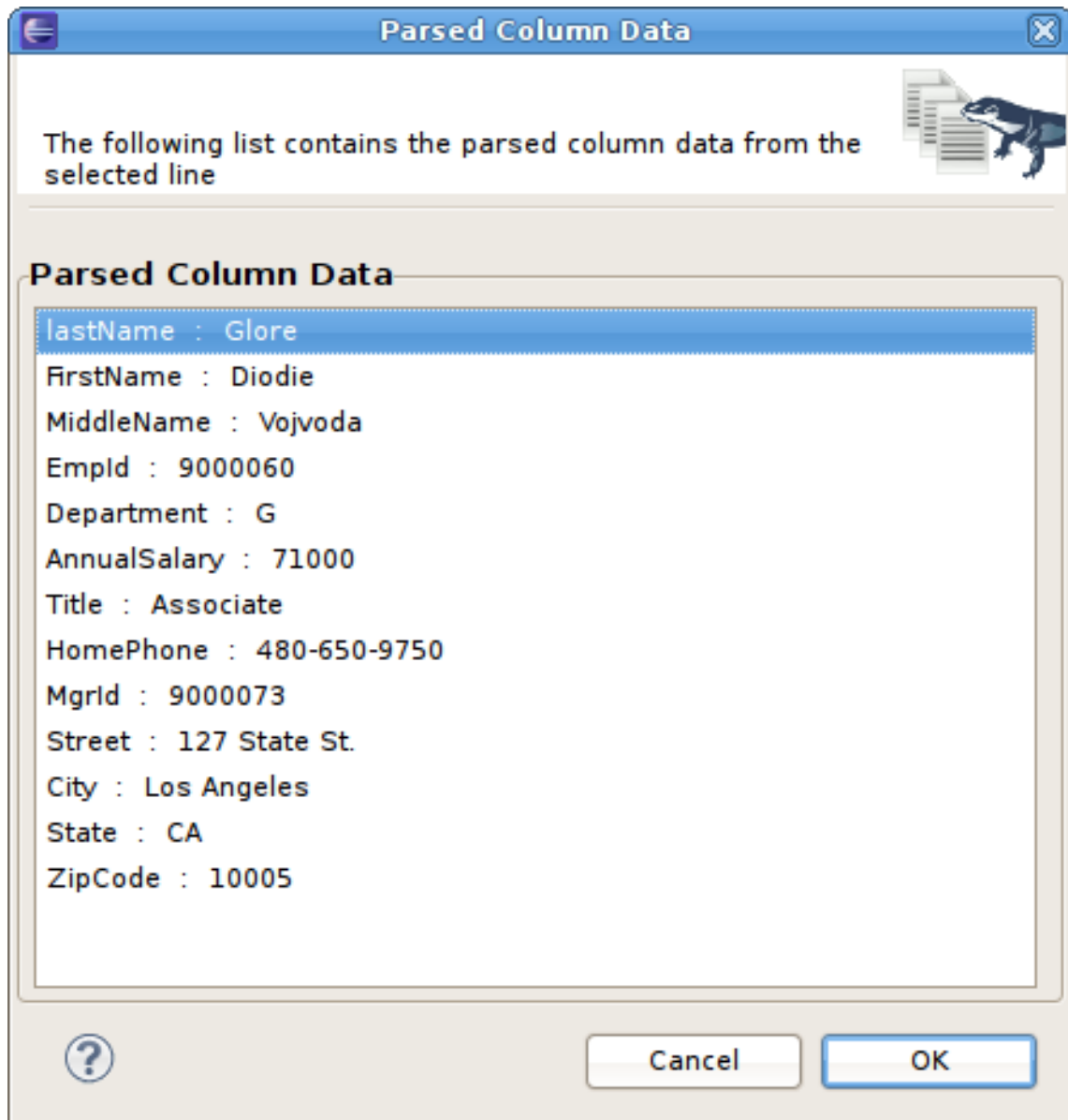


図9.14 Parse Column Data ダイアログ

6. Fixed Column Width (固定列幅) オプション - このインポーターの主な目的は、ユーザー定義のデータファイルをクエリーするために必要なトランスフォーマーションが含まれるビューテーブルを作成することです。このページには、Fixed Column Width オプションの下のパネル (Generated SQL Statement) に表示される生成された SQL ステートメントをカスタマイズするために使用できるオプションが示されます。ヘッダーオプション (ヘッダーの列名、ヘッダー行番号、および最初のデータ行番号) の指定、選択した行の解析、文字デリミターの変更、および **TEXTTABLE()** 関数オプションの編集を行います。**TEXTTABLE()** 関数の詳細は、Teiid User's Guide を参照してください。

ファイルヘッダーで列名が定義されていない場合や、カスタム列を編集または作成したい場合は、**ADD**、**DELETE**、**UP**、または **DOWN** を使用して SQL の列情報を管理できます。

左上のパネルにあるカーソルのポジションとテキストの長さの値を使用して、データファイルの列幅を判断することもできます。

このページの作業を終了したら **Next>** をクリックします。

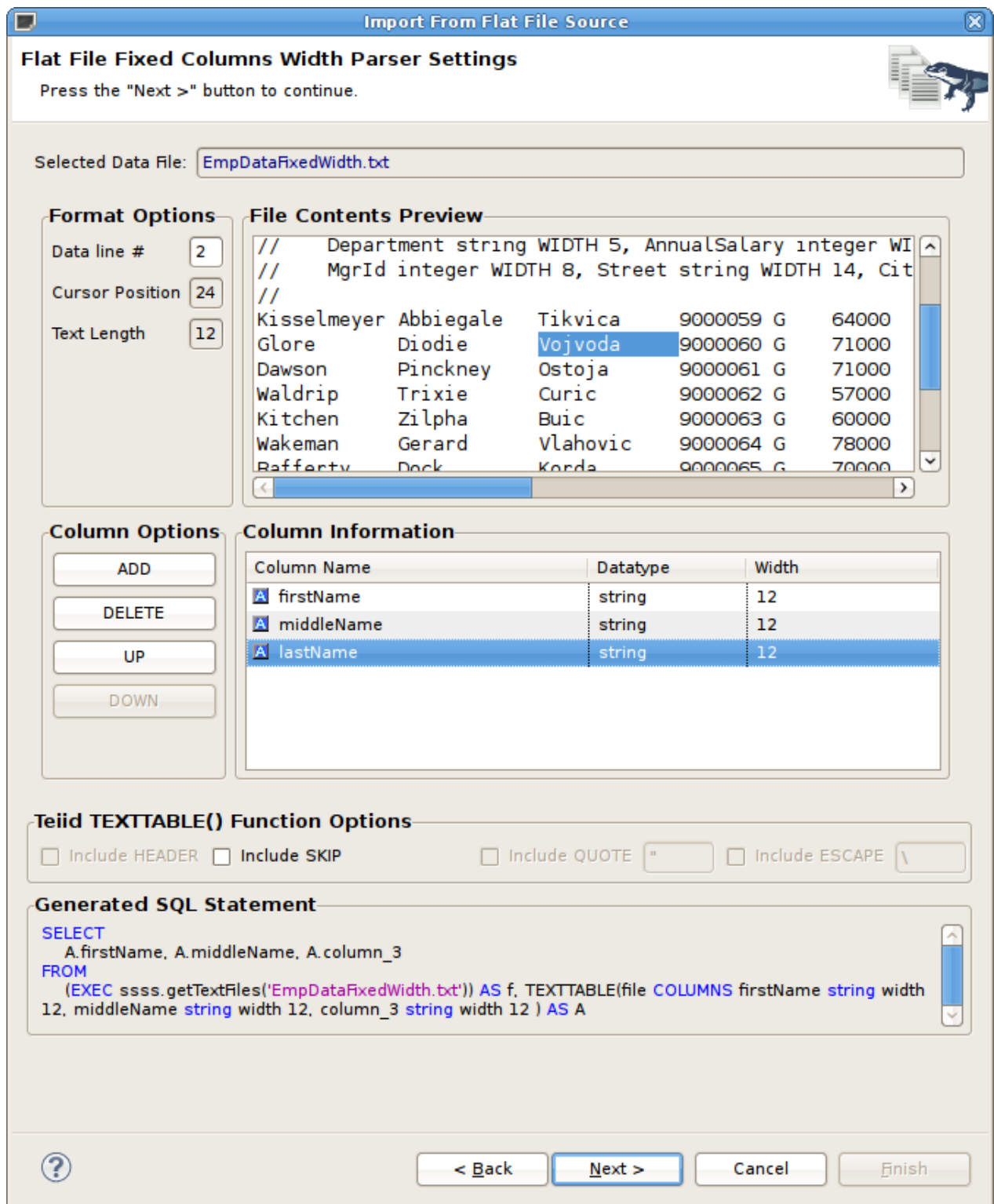


図9.15 Flat File Fixed Columns Width Options ページ

- View Model Definition ページで、新しいビューモデルを作成するフォルダーの場所を選択します。新しいビューテーブルに既存のモデルを選択することもできます。モデル名の有効性、モデルが存在するかどうか、およびモデルに `getTextFiles()` プロシージャがすでに含まれているかどうかを示す Model Status を確認してください。Finish をクリックしてモデルを生成し、ウィザードを終了します。



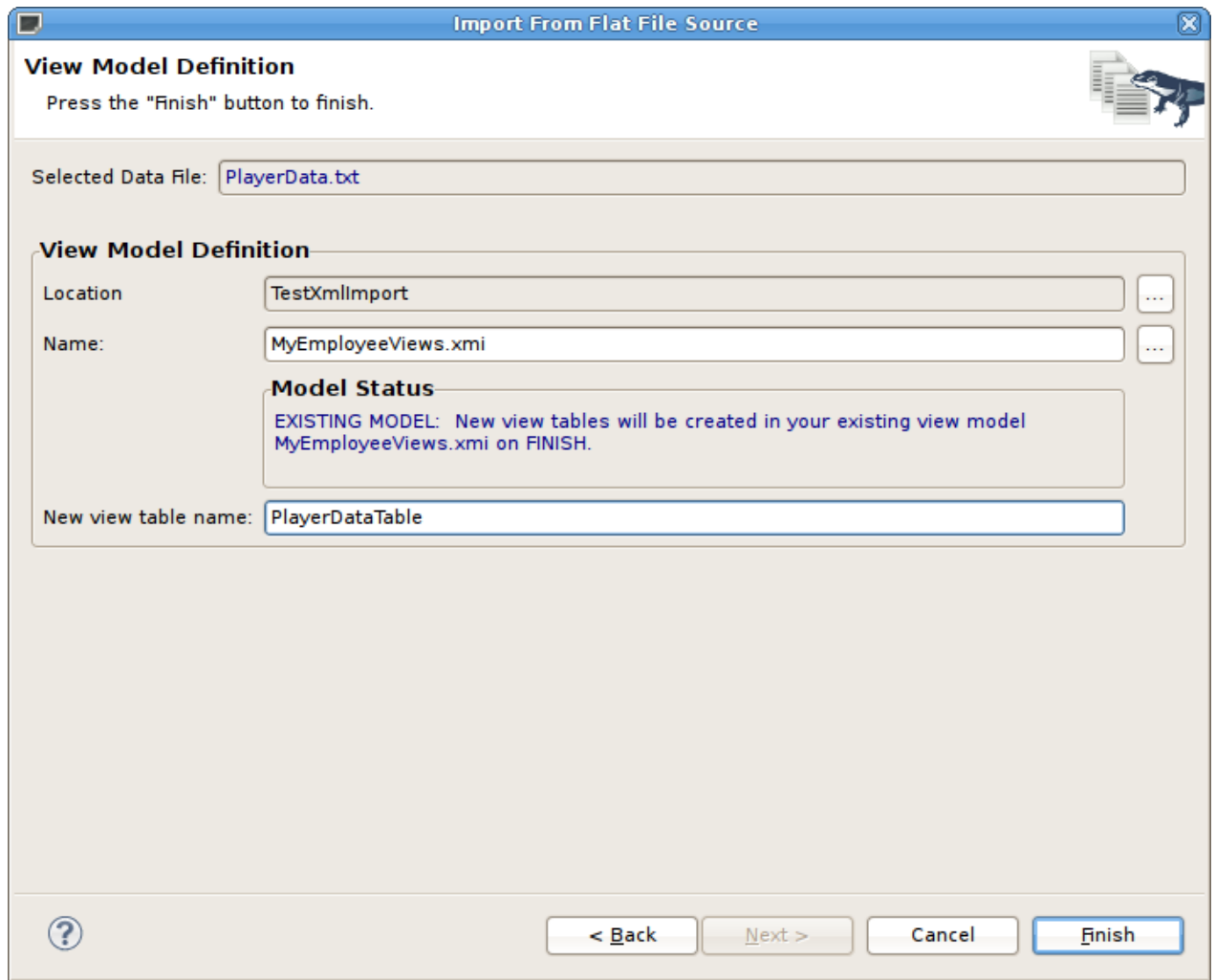


図9.16 View Model Definition ページ

インポートが終了すると、エディターでソースモデルが開かれ、`getTextFiles()` プロシージャが含まれる図が表示されます。

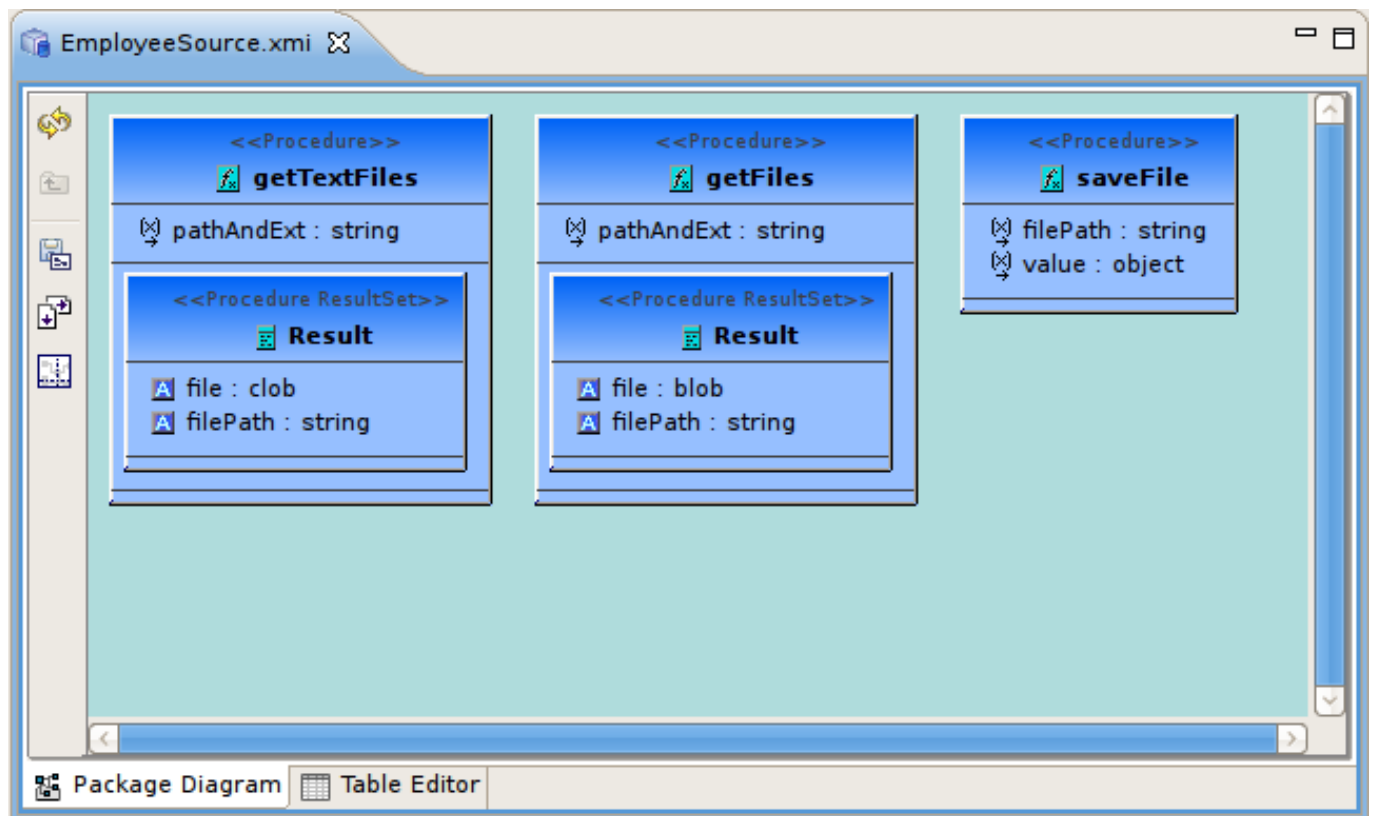


図9.17 生成されたフラットファイルプロシージャ

さらに、エディターでビューモデルが開かれ、上記の **getTextFiles** プロシージャおよび Teiid **TEXTTABLE()** 関数を使用してフラットファイルのデータにアクセスするために必要な完全な SQL が含まれる生成済みのビューテーブルが表示されます。以下の図は生成されたビューテーブルの例になります。

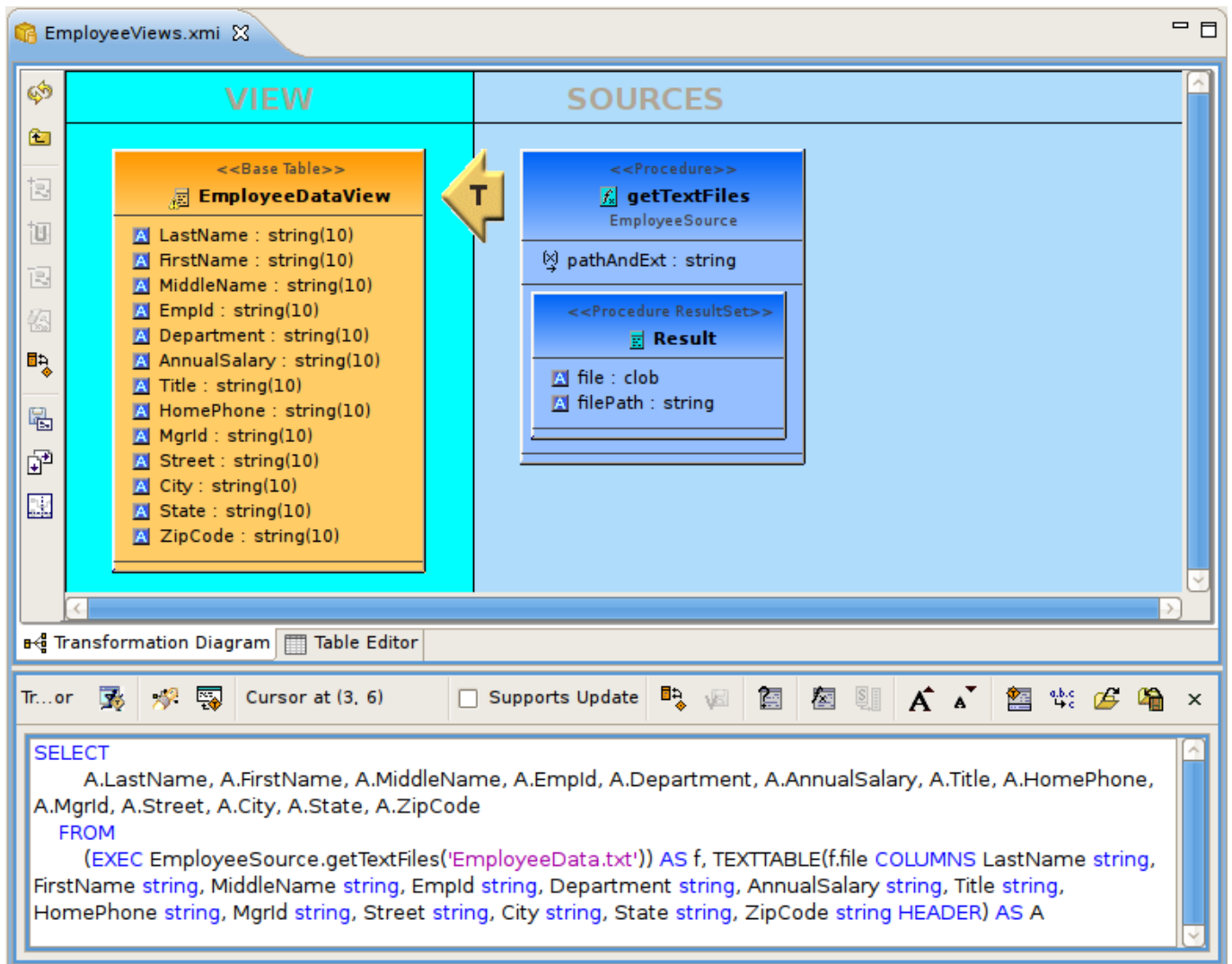


図9.18 生成されたフラットファイルビューテーブル

## 9.6. XML データファイルソースからのインポート

**JBoss Data Virtualization** はデータソースとしての XML ファイルをサポートします。これらのデータソースからインポートし、データを数分でクエリーするために必要なメタモデルを作成できます。以下の手順に従って、XML データソースの定義、XML データファイルの解析パラメーターの設定、必要な Teiid プロシージャが含まれるソースモデルの作成、および XML データファイルの列データを定義する SQL が含まれるビューテーブルの作成を行います。

Teiid Designer の JDBC、Salesforce、および WSDL インポーターと同様に、XML ファイルインポーターは特定のデータツール接続プロファイルの使用を基にします。

インポーターの結果には、`getTextFiles()` または `invokeHTTP()` プロシージャが含まれるソースモデルが含まれます。これらのプロシージャは JBoss Data Virtualization によってサポートされます。

インポーターは、選択した XML ソースファイルのビューテーブルが含まれる新しいビューモデルも作成します。ソースモデルからの `getTextFiles()` プロシージャが含まれる SQL トランスフォーメーションはビューテーブル内で生成され、Teiid `XMLTABLE()` 関数に必要な列定義およびパラメーターはデータファイルのクエリーに使用されます。また、新しいビューモデルを作成する代わりに既存のビューモデルの更新を選択することもできます。

**XMLTABLE** 関数は XQuery を使用して表形式出力を生成します。**XMLTABLE** 関数は暗黙的にはネストされたテーブルで、先行の FROM 句エントリーに関連付けることがあります。**XMLTABLE** は SQL/XML 2006 仕様の一部です。


```
XMLTABLE([<NSP>,] xquery-expression [<PASSING>] [COLUMNS <COLUMN>, ... ]]  
AS name
```

```
COLUMN := name (FOR ORDINALITY | (datatype [DEFAULT expression] [PATH  
string]))
```

Teiid Designer は各ビューテーブルの完全な SQL ステートメントを次の形式で構築します。

```
SELECT A.entryDate AS entryDate, A.internalAudit AS internalAudit FROM  
(EXEC CCC.getTextFiles('sample.xml')) AS f,  
XMLTABLE(XMLNAMESPACES('http://www.kaptest.com/schema/1.0/party' AS pty),  
'/pty:students/student' PASSING XMLPARSE(DOCUMENT f.file) COLUMNS entryDate  
FOR ORDINALITY, internalAudit string PATH '/internalAudit') AS A
```

XML データファイルソースからインポートするには、以下の手順に従います。

1. In Model Explorer, click **File > Import** action  in the toolbar or select a project, folder or model in the tree and choose Import...
2. インポートオプション **Teiid Designer > File Source (XML) >> Source and View Model** を選択し、**Next>** をクリックします。

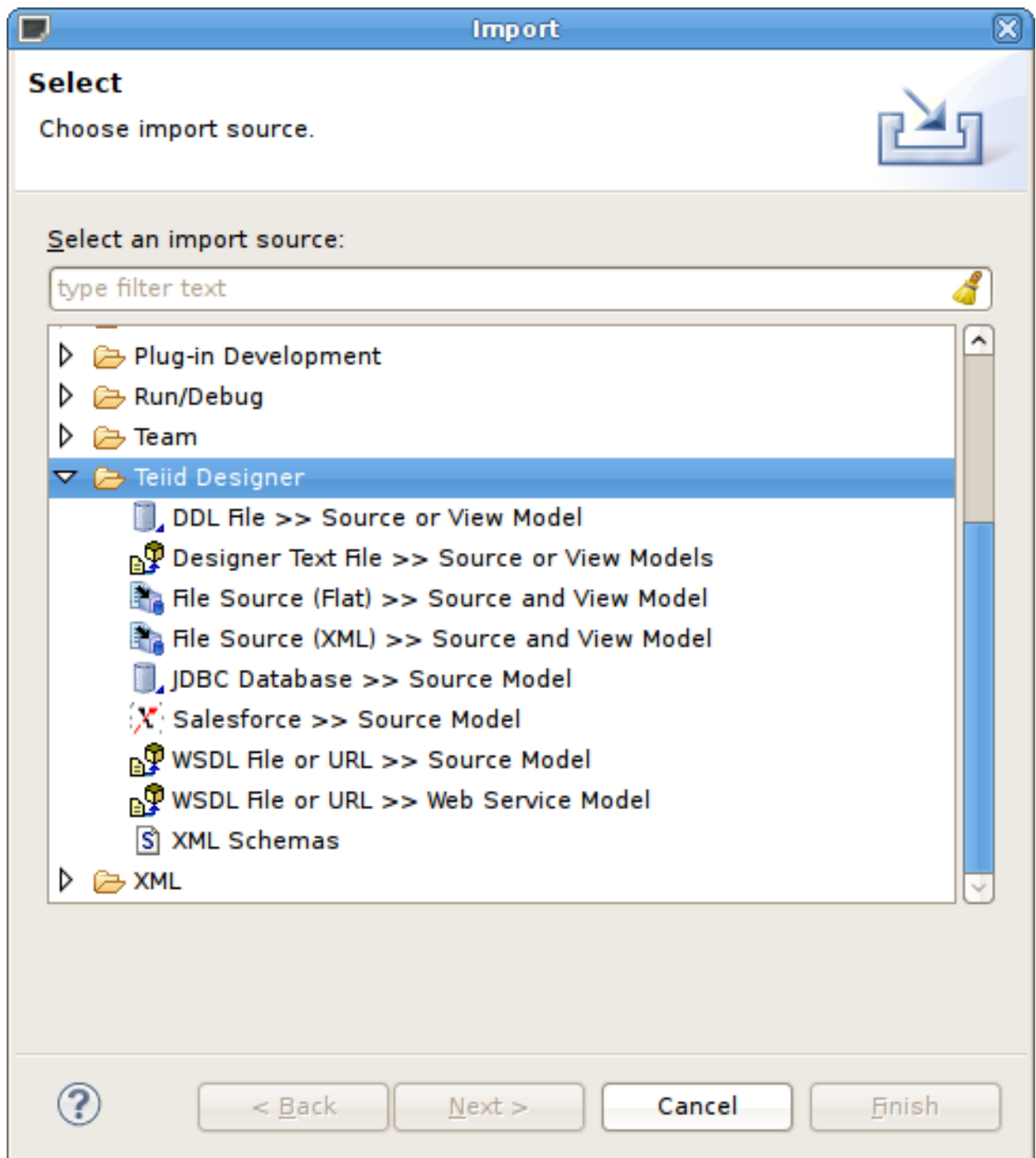


図9.19 XML ファイルソースからのインポート

3. ウィザードの次のページでは、XML ファイルがローカルまたはリモートであるかを指定する XML インポートモードを選択できます。上部の Description には、このウィザードが実行する操作の詳細が記載されています。**XML file on local file system** または **XML file via remote URL** を選択し、**Next>** をクリックします。

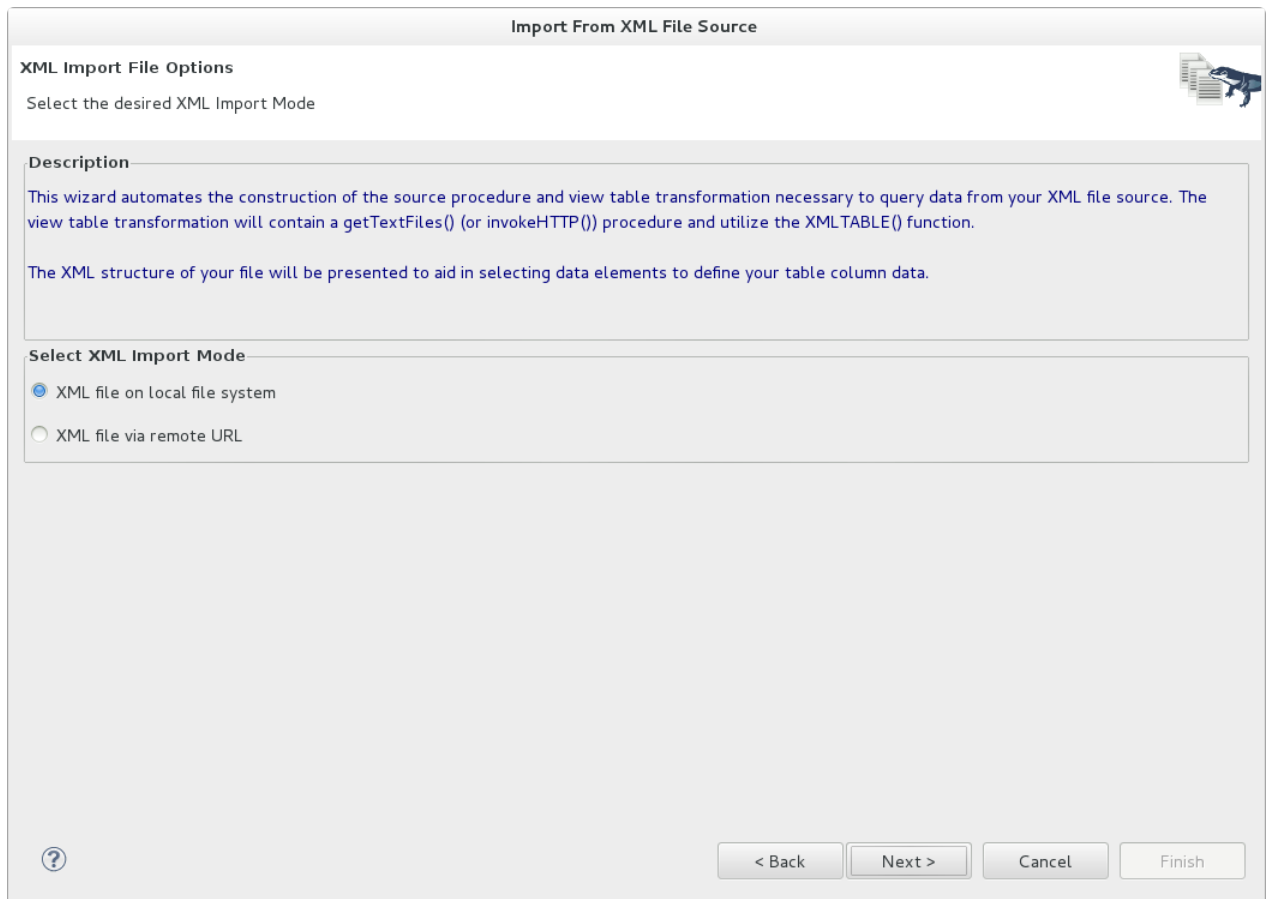


図9.20 XML Import File Options ページ

4. ドロップダウンメニューから既存または以前の接続プロファイルを選択するか、**New...** ボタンをクリックして **New Connection Profile** ダイアログを開始します。または、**Edit...** をクリックして選択前に既存の接続プロファイルを変更します。

新しい XML 接続プロファイルは、**XML File URL** または **REST Web Services URL** から作成できます。XML File URL オプションでは、ファイルまたは URL ベースの XML ドキュメントへの接続が有効になります。REST Web Service URL オプションでは、REST ベースの Web サービスへの接続が有効になります。

接続プロファイルを選択した後、接続プロファイルからの XML データファイルが **Available Data Files** パネルに表示されます。処理するデータファイルを選択します。データの読み出しと結果セットの返信に必要な SQL トランスフォーメーションが含まれるビューテーブルの構築には、このファイルからのデータとカスタムインポートオプションが使用されます。

最後に、ページの下部にある **Source Model Definition** に一意なソースモデル名を入力するか、**Browse** ボタンを使用して既存のソースモデルを選択します。モデル名の有効性、モデルが存在するかどうか、およびモデルに **getTextFiles()** プロシージャがすでに含まれているかどうかを示す **Model Status** を確認してください。この場合、ソースモデルもプロシージャも生成されません。

このページの作業を終了したら **Next>** をクリックします。

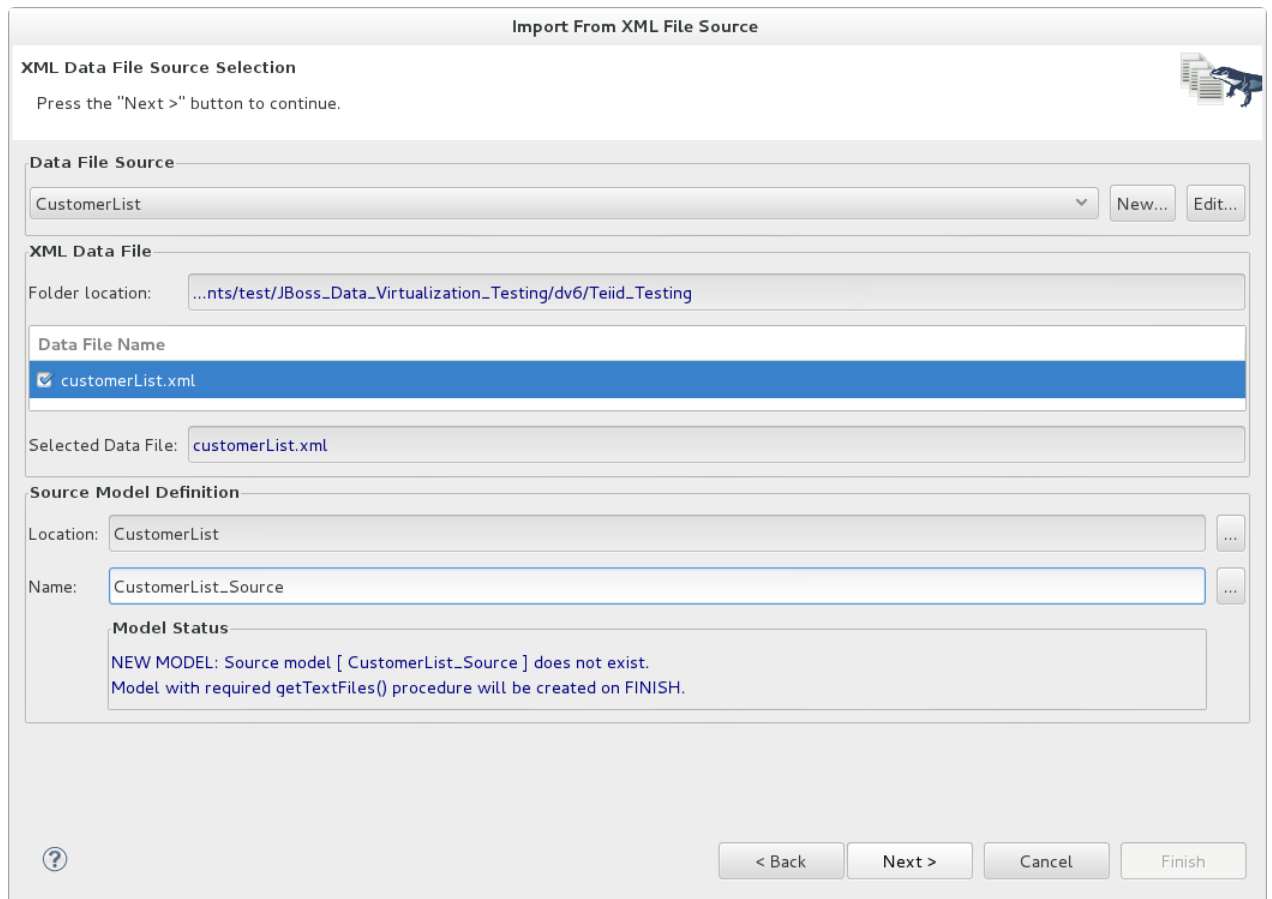


図9.21 XML Data File Source Selection ページ

- このインポーターの主な目的は、ユーザー定義のデータファイルをクエリーするために必要なトランスフォーメーションが含まれるビューテーブルを作成することです。このページには、下部のパネル (Generated SQL Statement) に表示される生成された SQL ステートメントをカスタマイズするために使用できるオプションが示されます。パネルには、XML ファイルの内容のツリービューと、中央にある Column Information パネルに表示される列エントリーの作成に使用するアクションやボタンが含まれます。

列を作成するには、ルート XML 要素を選択し、右クリックで **Set as root path** アクションを選択します。これにより、ルートパスの値が入力されます。次に、クエリーに含めたいツリーの列を選択します。**DELETE**、**UP**、または **DOWN** を使用して SQL の列情報を管理すると、カスタムの列を変更または作成できます。

列の **Path** プロパティの値は、定義されたルートパスを基にした選択された要素への相対パスになります。ルートパスが定義されていない場合は、パスはすべて絶対パスになります。各列エントリーにはデータタイプとオプションのデフォルト値が必要になります。**XMLTABLE()** 関数の詳細は Teiid User's Guide を参照してください。

このページの作業を終了したら **Next>** をクリックします。

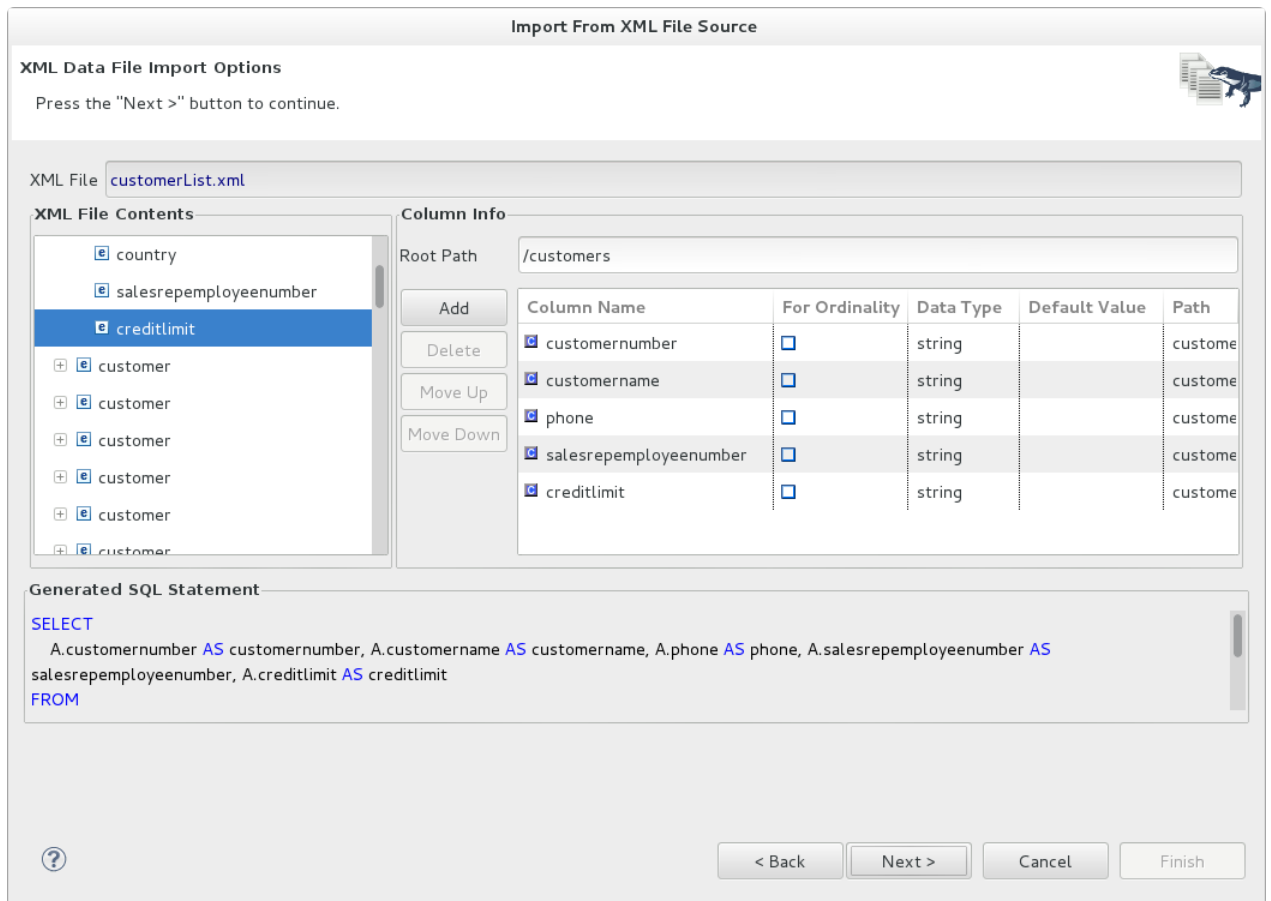


図9.22 XML File Delimited Columns Options ページ

6. **View Model Definition** ページで、新しいビューモデルを作成するフォルダーの場所を選択します。新しいビューテーブルに既存のモデルを選択することもできます。モデル名の有効性、モデルが存在するかどうか、およびモデルに `getTextFiles()` プロシージャがすでに含まれているかどうかを示す **Model Status** を確認してください。**Finish** をクリックしてモデルを生成し、ウィザードを終了します。



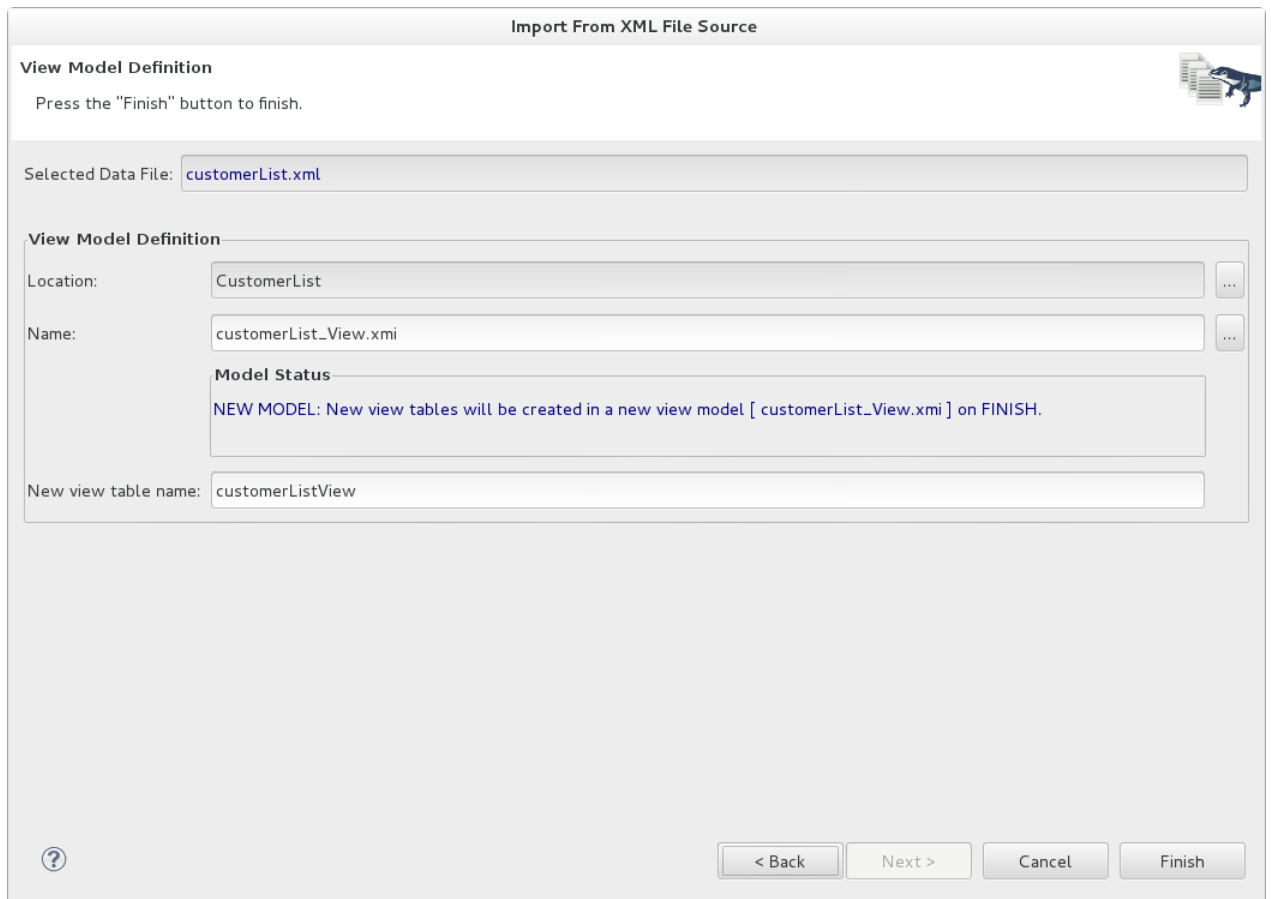


図9.23 View Model Definition ページ

## 9.7. Salesforce からのインポート

Salesforce 接続からリレーショナルソースモデルを作成するには、以下の手順に従います。

### 注記

データベース接続 URL 情報およびスキーマで提供される詳細によっては、ステップ 5 から 7 までを行う必要がない場合があります。

1. Model Explorer でツールバーの **File > Import** アクションをクリックするか、ツリーのプロジェクト、フォルダー、またはモデルを選択し、**Import...** をクリックします。
2. インポートオプション **Teiid Designer > Salesforce >> Source Model** を選択し、**Next>** をクリックします。
3. ドロップダウンメニューから既存または以前の接続プロファイルを選択するか、**New...** ボタンをクリックして **New Connection Profile** ダイアログを開始します。または、**Edit...** をクリックして選択前に既存の接続プロファイルを変更します。Connection Profile 選択リストは、Salesforce 接続プロファイルでのみ入力されます。

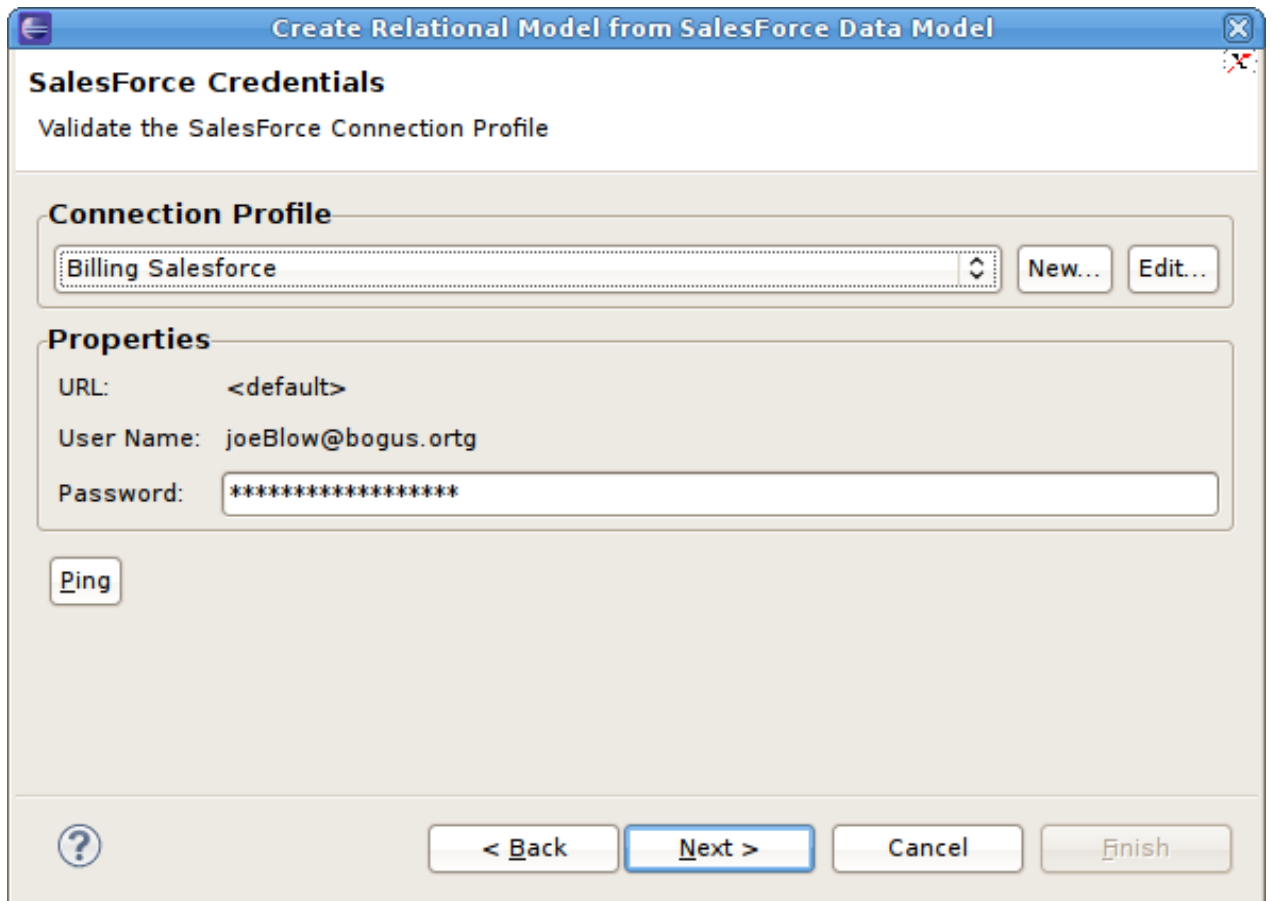


図9.24 Select Salesforce Credentials ダイアログ

4. 接続プロファイルを選択した後、パスワードを入力します (入力済みでない場合)。**Next>>** をクリックし、Salesforce Objects の選択ページを表示します。

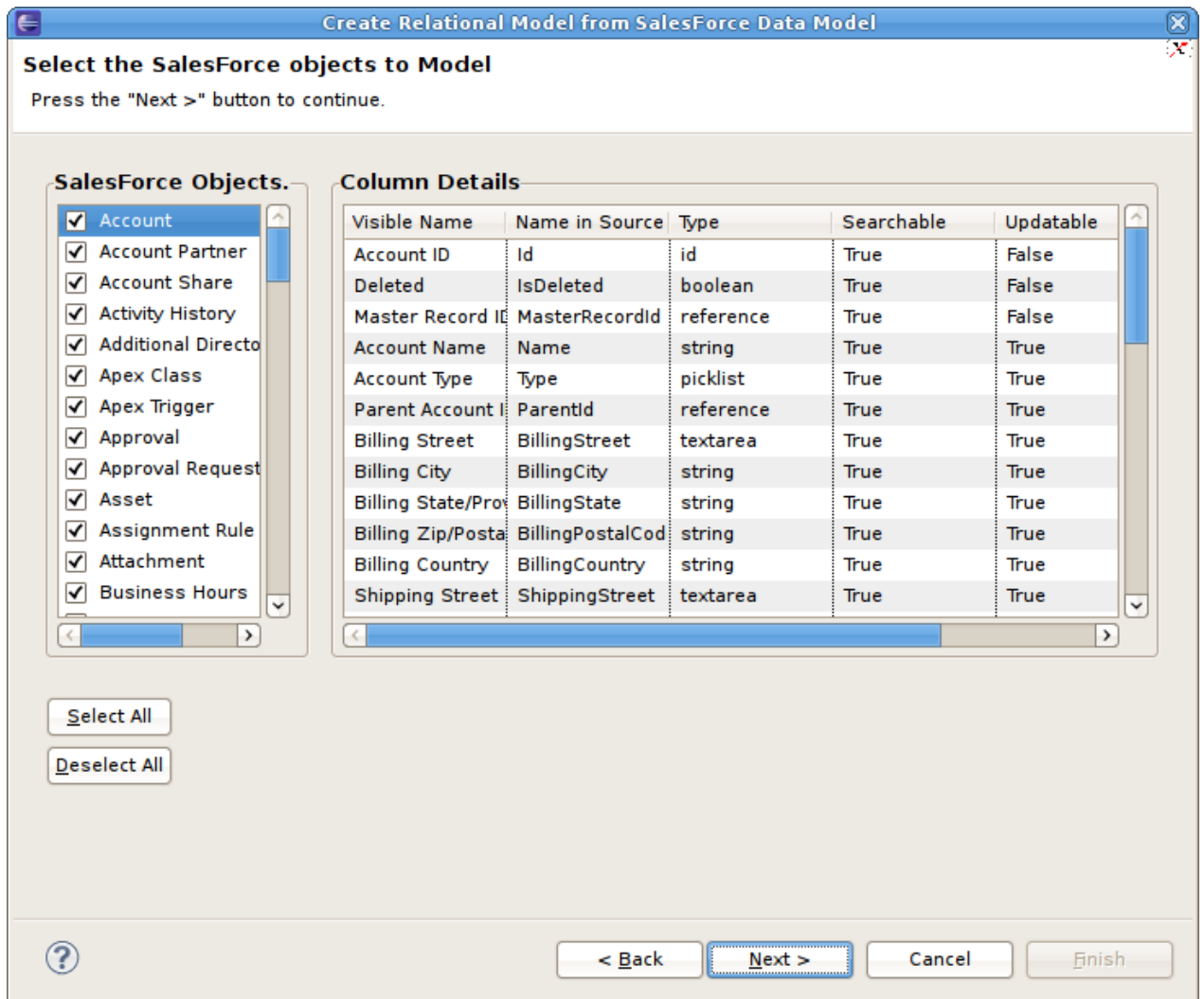


図9.25 Select Salesforce Objects ダイアログ

5. **Target Model Selection** ページで、生成されたモデルのフォルダーの場所と一意なモデル名を指定し、必要なインポートオプションを選択します。**Next>** (または有効な場合は **Finish**) をクリックします。

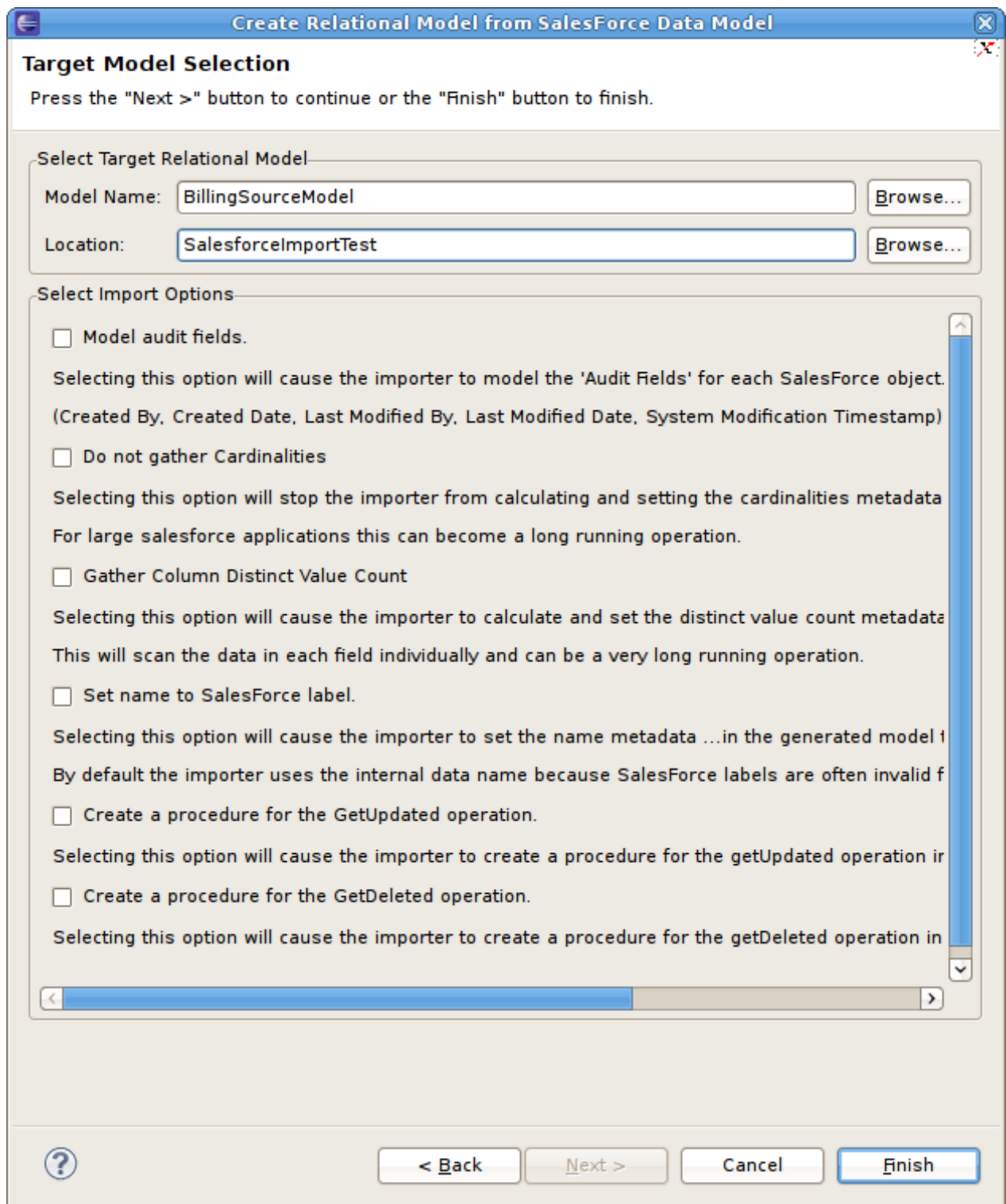


図9.26 Target Model Selection ダイアログ

6. 既存のリレーショナルモデルを更新している場合は、次のページは **Review Model Updates** ページになります。 **Finish** をクリックしてモデルとテーブルを作成します。

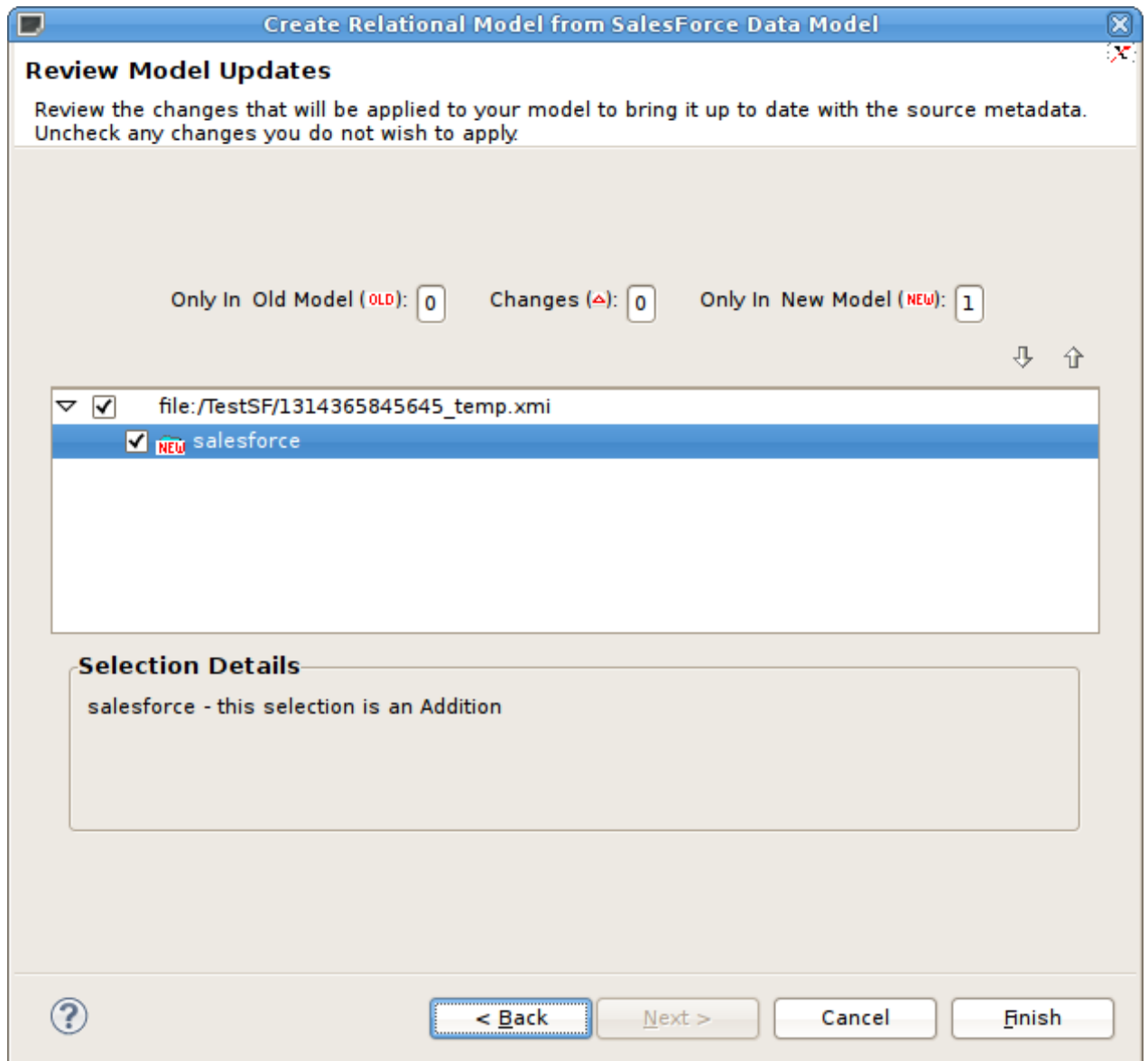


図9.27 Review Model Updates ダイアログ

7. 終了したら、新しいまたは変更されたリレーショナルモデルのパッケージ図が表示され、新しいテーブルが表示されます。

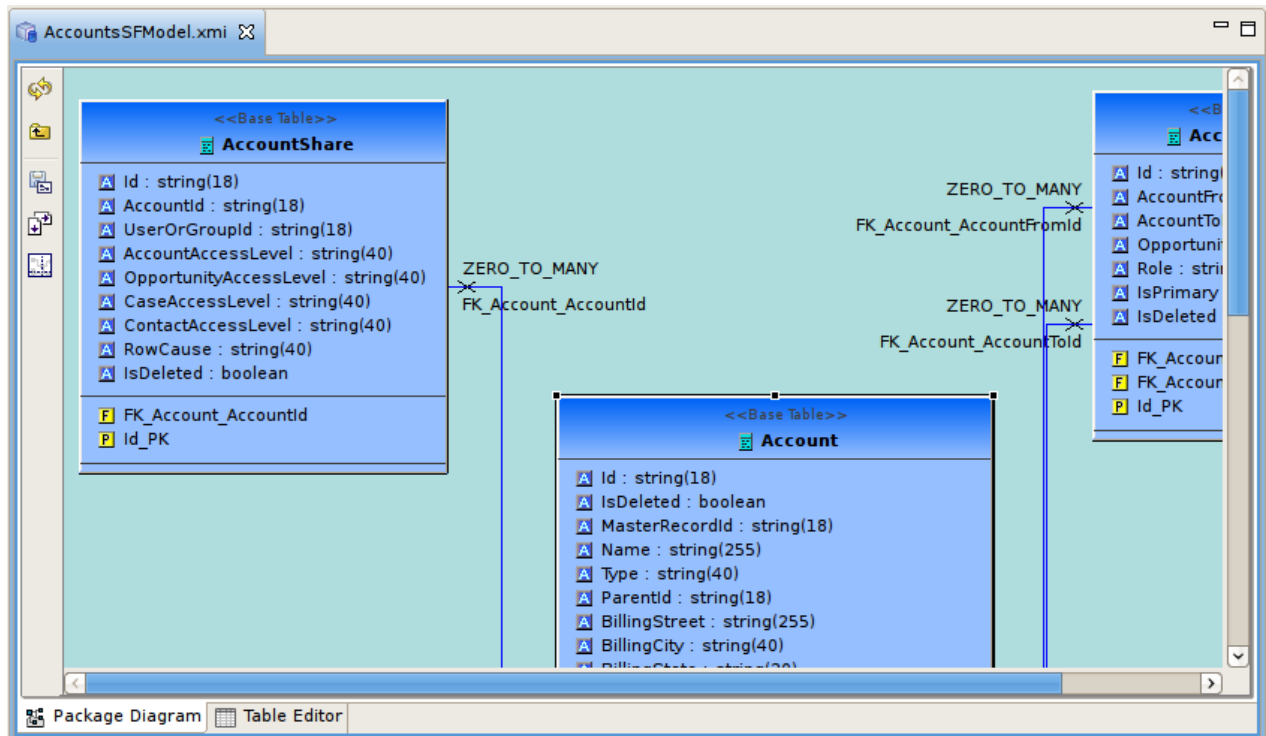


図9.28 New Salesforce Tables ダイアグラム

## 9.8. テキストファイルからのメタデータのインポート

### 9.8.1. テキストファイルからのメタデータのインポート

Teiid Designer にはカンマ区切りのテキストファイルメタデータをモデルに解析する複数のインポートオプションがあります。これは、**Import > Teiid Designer > Designer Text File >> Source or View Models** と選択して行います。

1. Teiid Designer で **File > Import** と選択し、**Import...** をクリックします。
2. インポートオプション **Teiid Designer > Designer Text File >> Source or View Models** を選択し、**Next>** をクリックします。
3. 以下のドロップダウンメニューからインポートタイプを選択します。

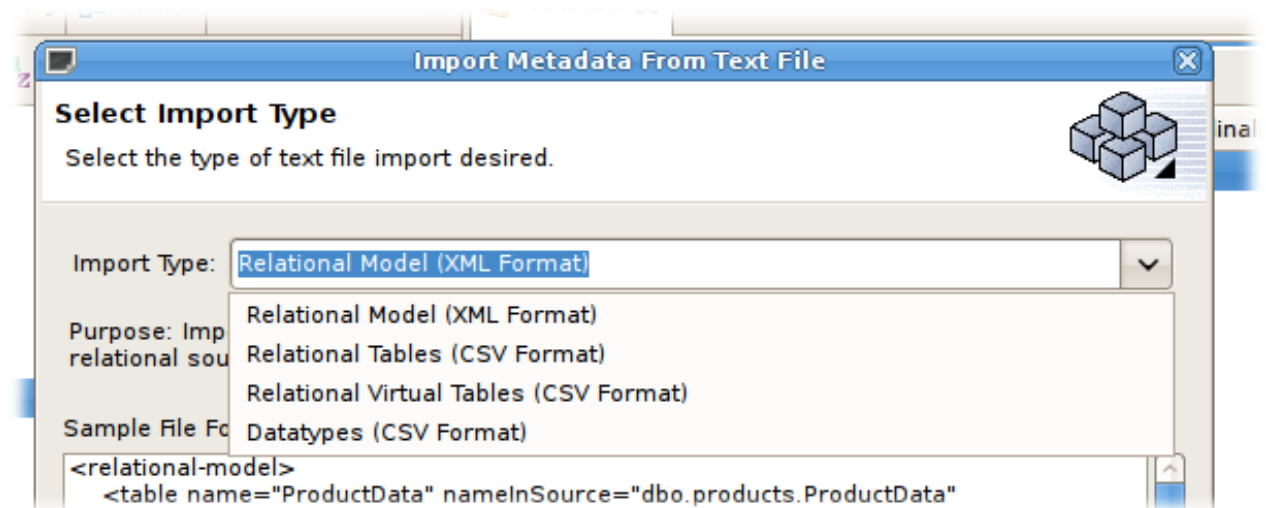


図9.29 Import ウィザード

これらの手順は以下の各タイプに必要です。

- ※ リレーショナルモデルテキストインポート
- ※ リレーショナルテーブルテキストインポート
- ※ 仮想テーブルテキストインポート
- ※ データタイプテーブルテキストインポート

### 9.8.2. リレーショナルモデルのインポート (XML 形式)

インポートした XML テキストファイルメタデータからリレーショナルテーブルを作成するには、「テキストファイルからのメタデータのインポート」に記載されている 1 から 3 までの手順を行った後、以下の手順を実行します。

1. **Relational Model (XML Format)** インポートタイプを選択した後、**Next >** をクリックします。

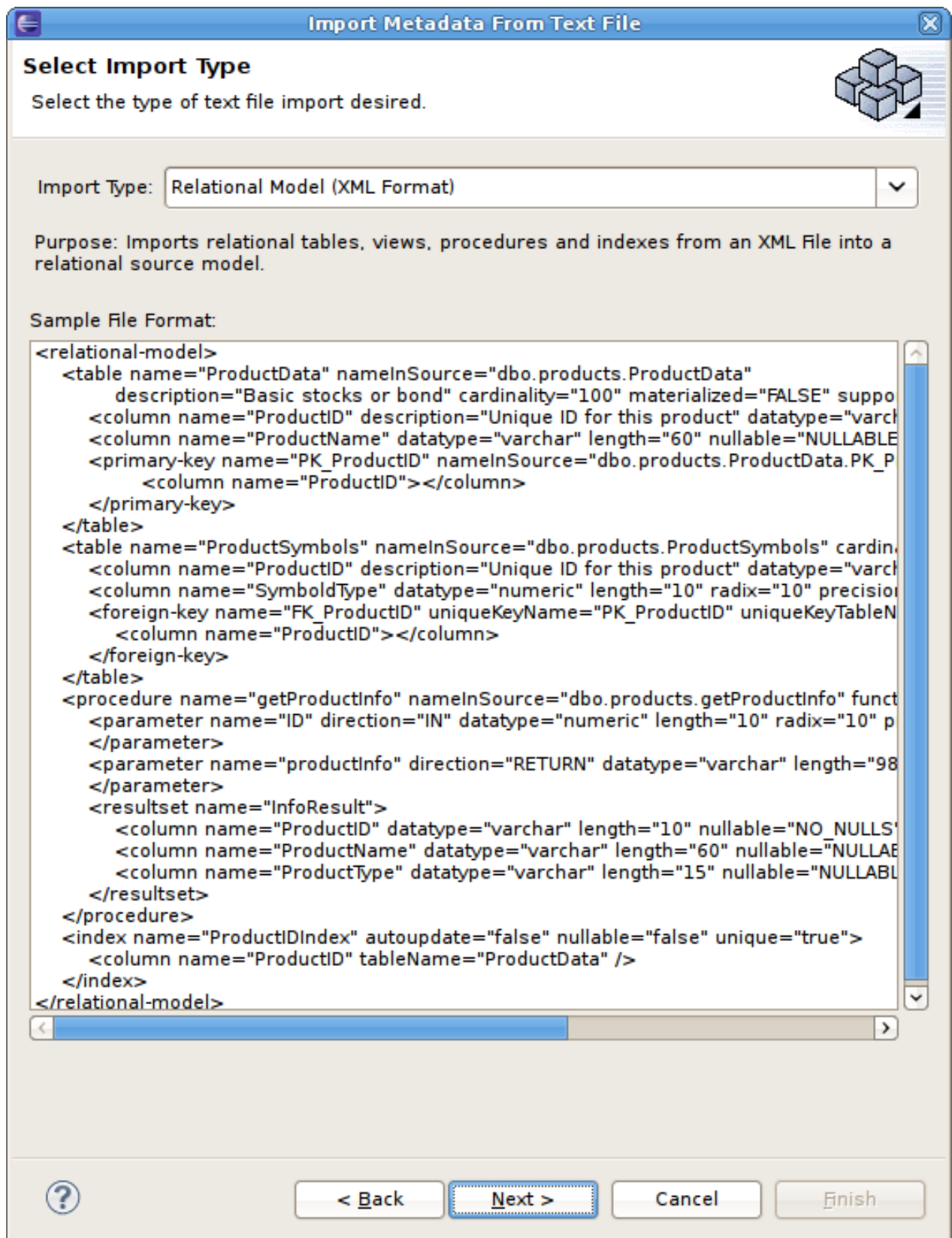


図9.30 インポートタイプの選択 - リレーショナルモデル (XML 形式)

- 次のページでは、**Browse...** ボタンを使用してローカルファイルシステムの XML ファイルを選択します。2つ目の **Browse...** ボタンを使用してインポートされたリレーショナルオブジェクトを追加するモデルを選択します。ダイアログでは、既存のリレーショナルモデルを選択したり、新しいモデルを作成したりできます。選択した XML ファイルの内容は File Contents ビューアーに表示されます。**Finish** をクリックします。



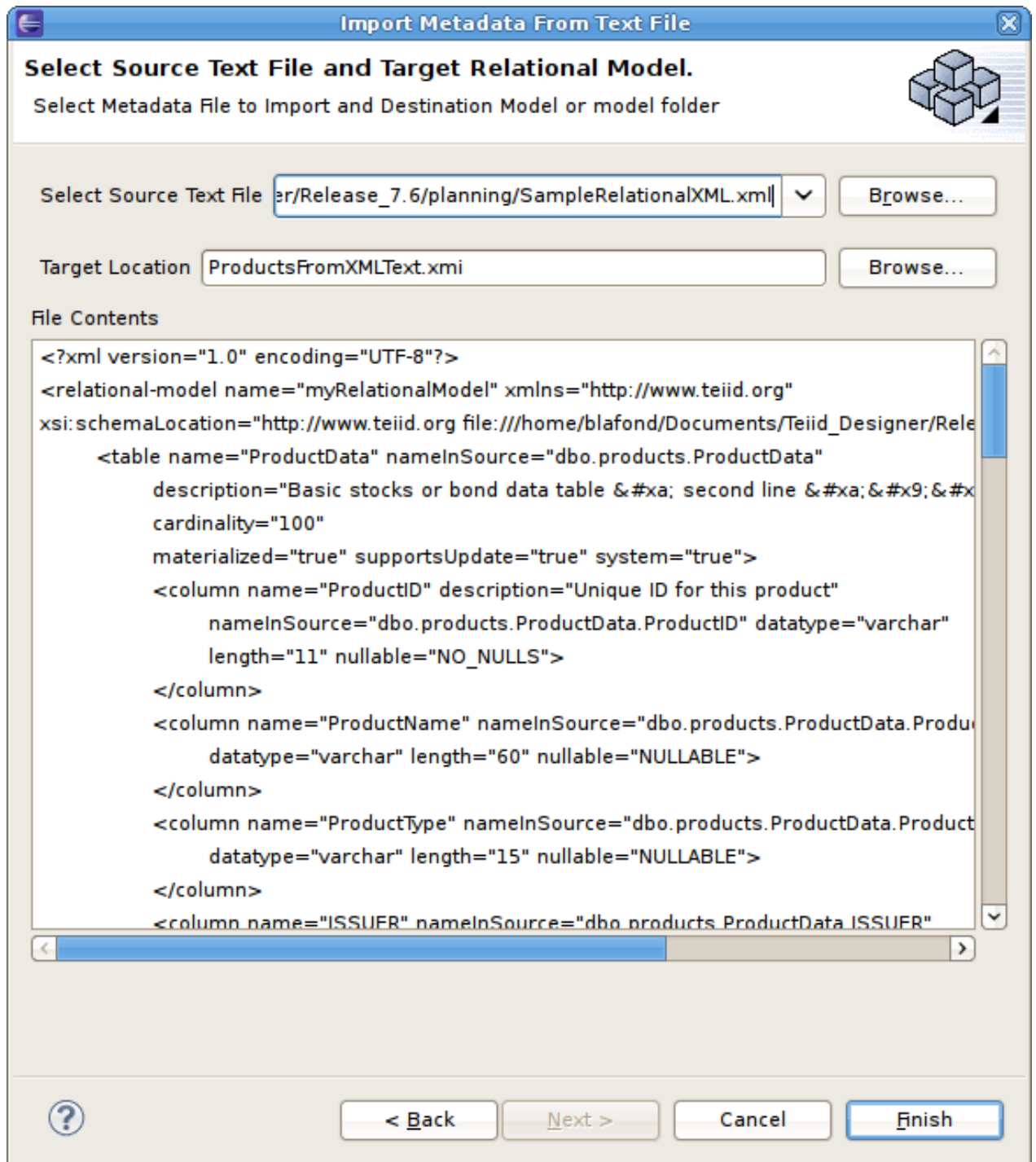


図9.31 Select Source Text File and Target Relational Model ページ

- ターゲットのモデルにインポートしたオブジェクトと競合する名前の子 (テーブル、ビュー、プロシージャ) が含まれている場合はダイアログが表示され、特定の既存オブジェクトの置き換え、同じ名前のオブジェクトの新規作成、またはインポートのキャンセルをオプションで選択できます。

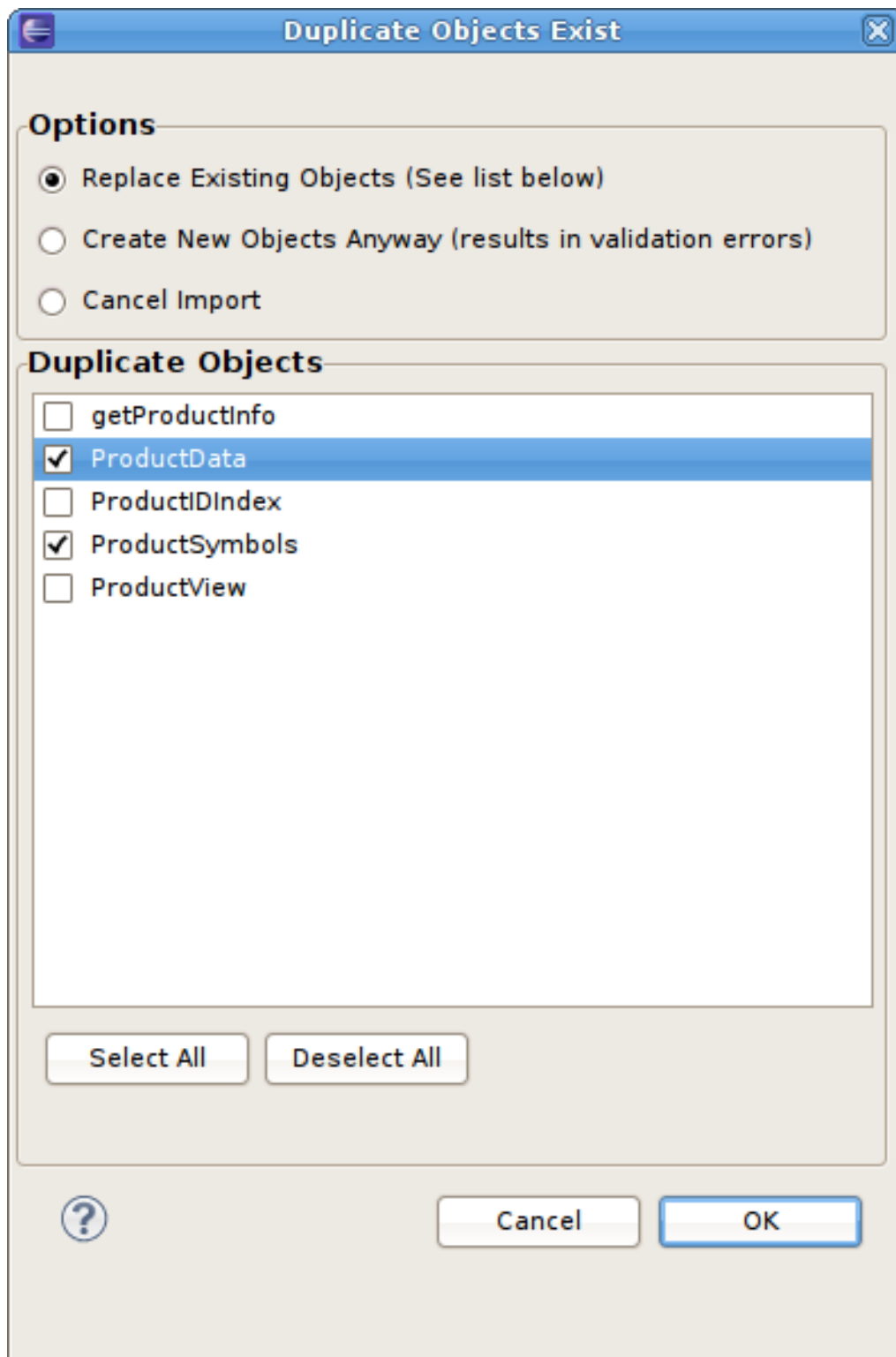


図9.32 Duplicate Objects ダイアログ

### 9.8.3. リレーショナルモデルのインポート (CSV 形式)

インポートしたテキストファイルメタデータからリレーショナルテーブルを作成するには、「テキストファイルからのメタデータのインポート」に記載されている 1 から 3 までの手順を行った後、以下の手順を実行します。

1. **Relational Tables (CSV Format)** インポートタイプを選択した後、**Next >** をクリックします。

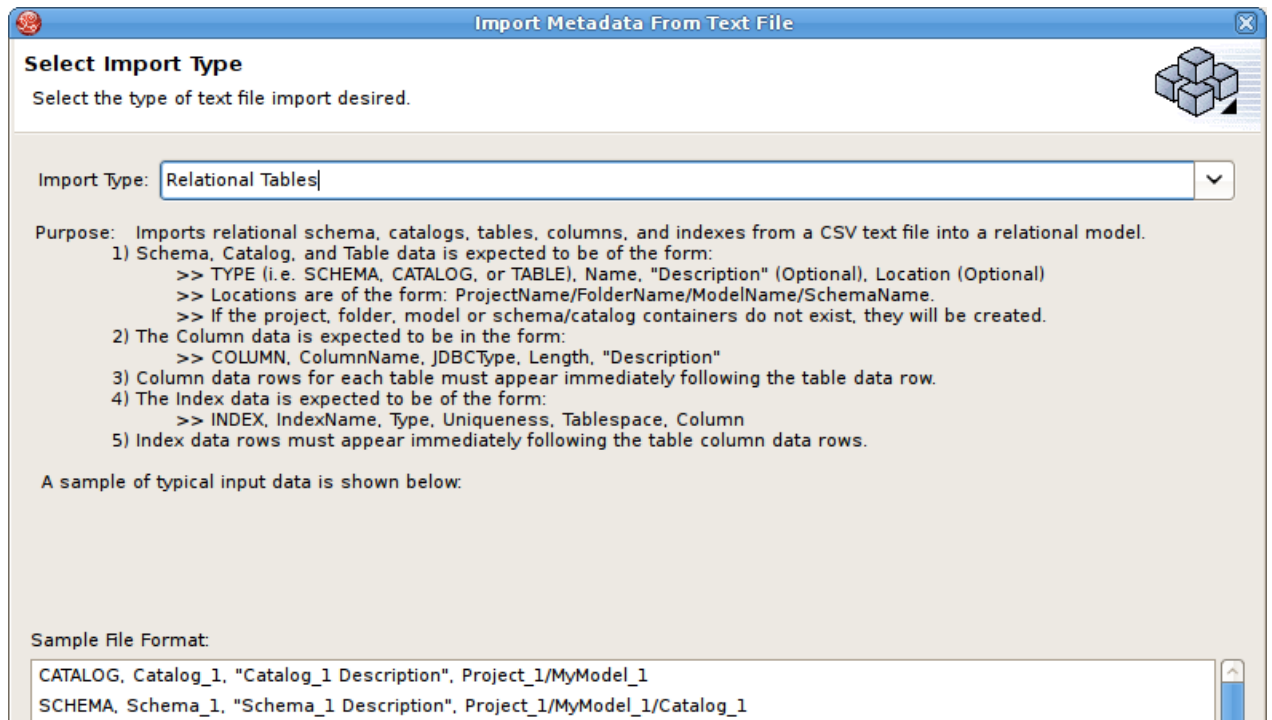


図9.33 インポートタイプの選択 - リレーショナルテーブル (CSV 形式)

2. 次のページでは、以前のページの仕様にフォーマットされたメタデータが含まれるソーステキストファイルを提供する必要があります。

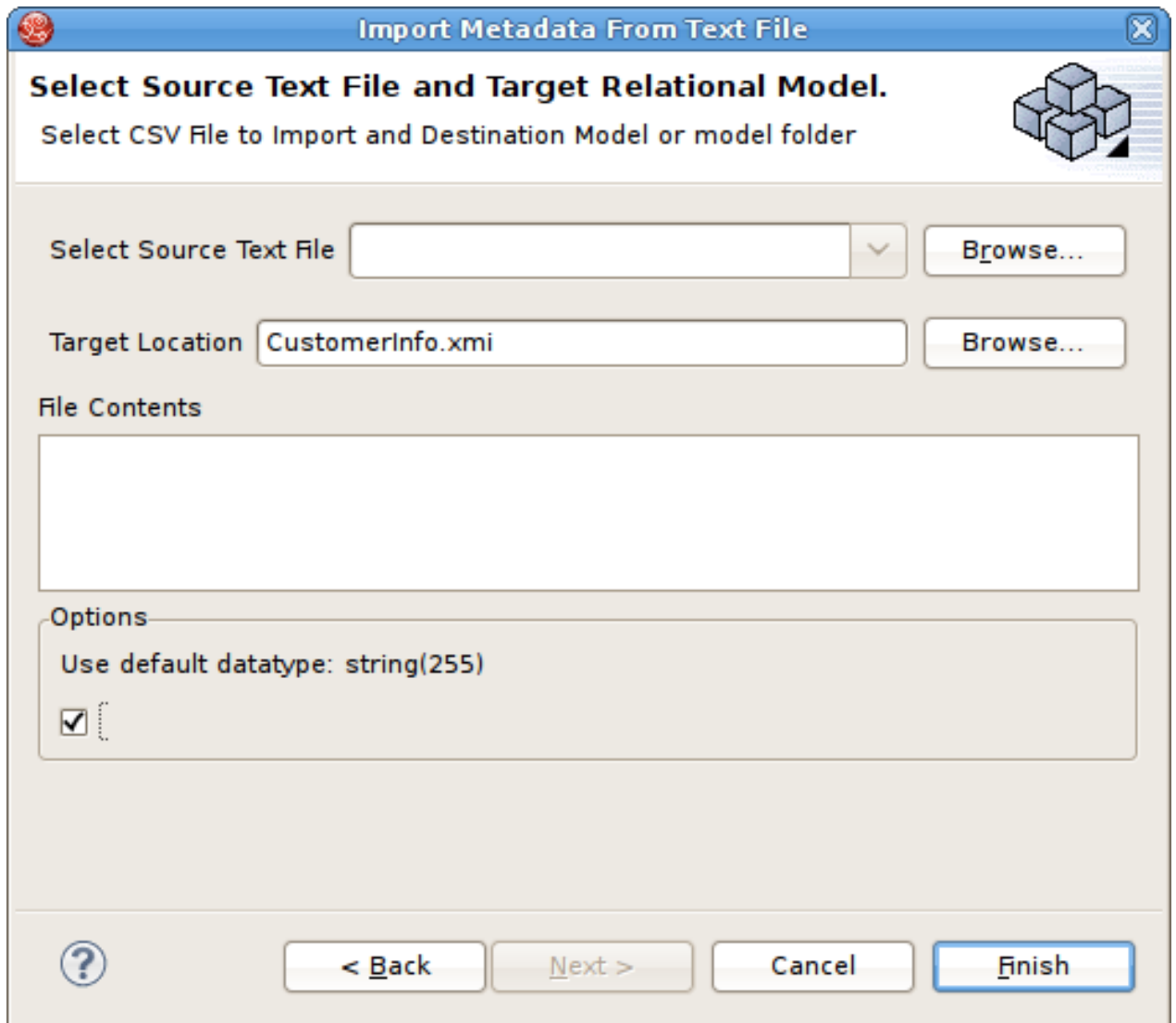


図9.34 Select Source Text File and Target Relational Model

3. **Browse...** ボタンを使用して **Relational Model Selector** ダイアログを開き、新しいリレーショナルコンポーネントの目的の場所として既存のリレーショナルモデルを選択します。ワークスペースからリレーショナルモデルを選択するか、一意の名前を指定して新しいモデルを作成します。
4. 追加のオプションを選択し、**Finish** をクリックします。

#### 9.8.4. リレーショナルビューテーブルのインポート (CSV 形式)

インポートしたテキストファイルメタデータからリレーショナル仮想テーブルを作成するには、「テキストファイルからのメタデータのインポート」に記載されている 1 から 3 までの手順を行った後、以下の手順を実行します。

1. **Relational Virtual Tables (CSV Format)** インポートタイプを選択した後、**Next >** をクリックします。

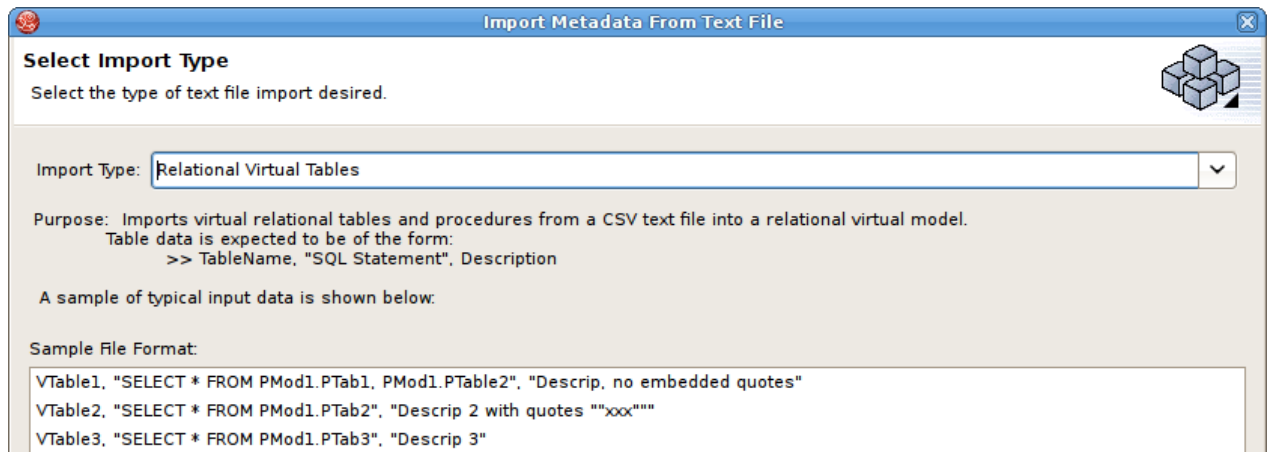


図9.35 インポートタイプの選択 - リレーショナル仮想テーブル (CSV 形式)

- 次のページでは、以前のページの仕様にフォーマットされたメタデータが含まれるソーステキストファイルを提供する必要があります。

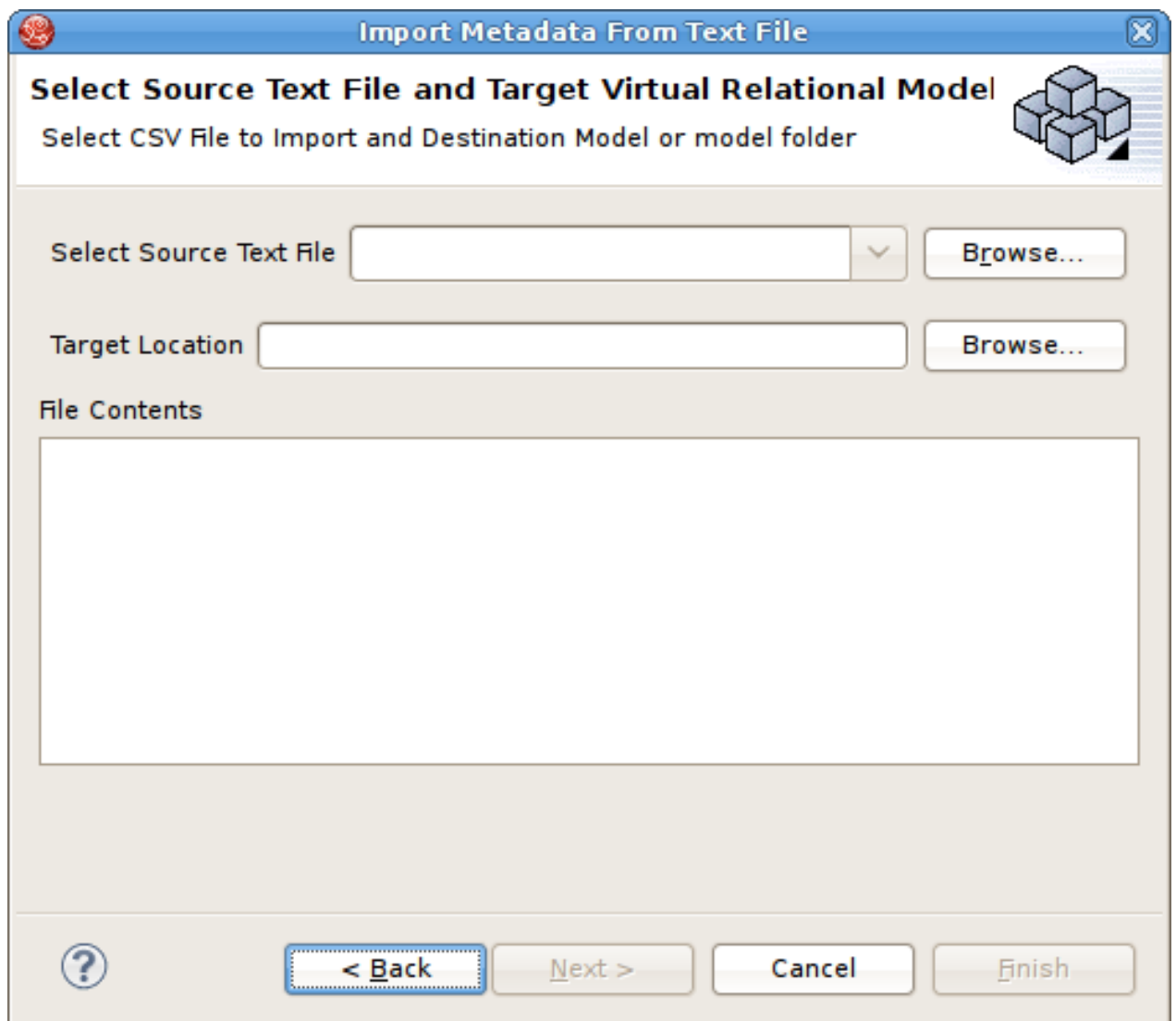


図9.36 Select Source Text File and Target Virtual Relational Model

- Browse...** ボタンを使用して **Virtual Model Selector** ダイアログを開き、新しいモデルコンポーネントの目的の場所として既存のリレーショナル仮想モデルを選択します。ワークスペースから仮想リレーショナルモデルを選択するか、一意の名前を指定して新しいモデルを作成します。

4. **Finish** をクリックします。

### 9.8.5. データタイプのインポート (CSV 形式)

インポートしたテキストファイルメタデータからデータタイプを作成するには、「テキストファイルからのメタデータのインポート」に記載されている 1 から 3 までの手順を行った後、以下の手順を実行します。

1. **Datatypes (CSV Format) import** タイプを選択した後、**Next >** をクリックします。

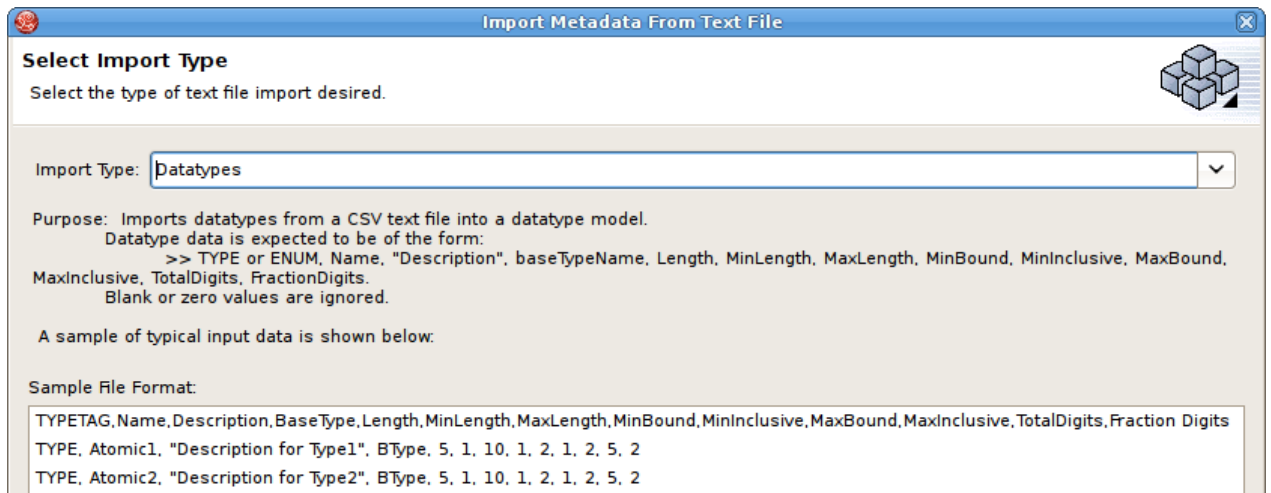


図9.37 インポートタイプの選択 - データタイプ (CSV 形式)

2. 次のページでは、以前のページの仕様にフォーマットされたメタデータが含まれるソーステキストファイルを提供する必要があります。

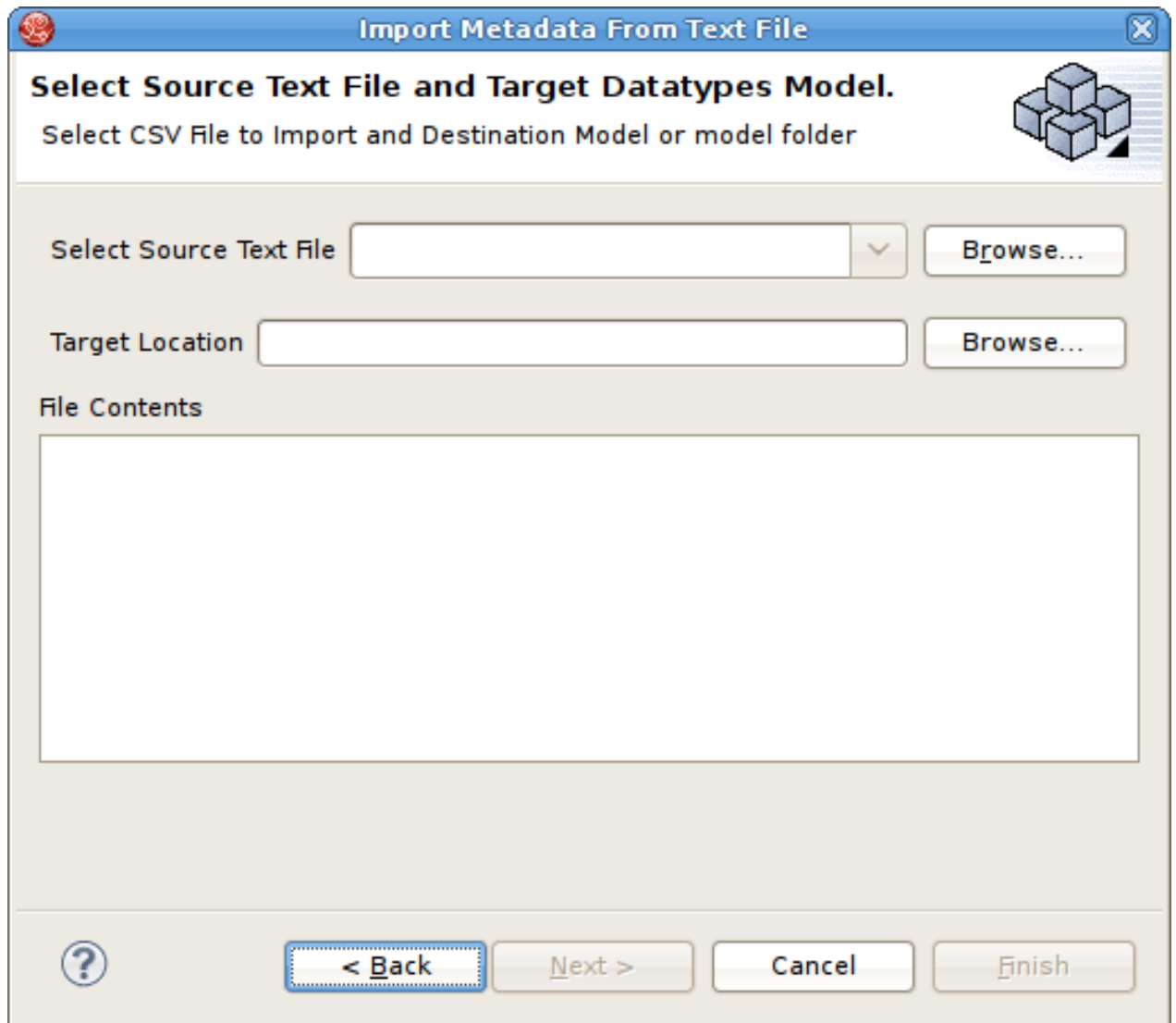


図9.38 Select Source Text File and Datatypes Model

3. **Browse...** ボタンを使用して **Datatypes Model Selector** ダイアログを開き、新しいモデルコンポーネントの目的の場所として既存のデータタイプモデルを選択します。ワークスペースからワークタイプモデルを選択するか、一意の名前を指定して新しいモデルを作成します。
4. 追加のオプションを選択し、**Finish** をクリックします。

## 9.9. WSDL のリレーショナルモデルへのインポート

### 9.9.1. WSDL のリレーショナルモデルへのインポート

WSDL ファイル (ローカルまたは URL) を、WSDL のスキーマ定義を通じて定義されるリクエストおよび応答 Web サービス構造を表す、クエリー可能なリレーショナルプロシージャーに変換できます。このウィザードを使用するには、**Import...** アクションをクリックした後に **Teiid Designer > WSDL File or URL >> Source and View Model (SOAP)** とオプションを選択します。Web サービス接続プロファイルは、ワークスペースの WSDL ファイルまたは URL によって定義されます。Teiid Designer は WSDL を解釈し、関連する XML スキーマファイルまたは依存する XML スキーマファイルを見つけ、物理モデルを生成してサービスを呼び出します。さらに、プロシージャーが含まれる仮想モデルを生成し、サービスメッセージとして宣言された XML を構築および解析します。

WSDL からリレーショナルモデルを作成するには、以下の手順に従います。

1. Model Explorer でツールバーの **File > Import** アクションをクリックするか、ツリーのプロジェクト、フォルダー、またはモデルを選択し、**Import...** をクリックします。
2. インポートオプションを **Teiid Designer > WSDL File or URL >> Source and View Model (SOAP)** と選択し、**Next>** をクリックします。
3. 次のページでリストから既存の Web サービス接続プロファイルを選択するか、**New** ボタンをクリックして新しいプロファイルを作成します。

図9.39 WSDL ソースの選択


4. 個別の Web サービス演算を選択し、モデル化します。このページのデフォルト動作では、ツリーで利用可能な演算がすべて選択されます。モデル化されない場合は演算の選択を解除できます。Selection Details パネルには、入出力メッセージの名前や演算によって発生した障害など、演算に関する静的な情報が表示されます。

Next > をクリックします。

5. 次のページは **Model Definition** というタイトルで、モデルの場所 (フォルダーまたはプロジェクトなど) とソースおよびビューモデル両方の有効なモデル名が必要になります。**Browse...** ボタンを使用して既存のフォルダーまたはモデルを選択します。すべての情報が定義されたら **Next>** をクリックします。



Create Relational Model from Web Service

**Models Definition**  
All inputs OK. Click 'Next>' to define custom procedures. 

**Source Model Definition**

Location:  ...

Name:  ...

Status:  
Source model CountryInfoService.xmi does not exist and will be created and contain the required invoke() web service procedure.

**View Model Definition**

Location:  ...

Name:  ...

Status:  
View model CountryInfoServiceView.xmi does not exist and will be created and contain your generated procedures.

**Procedure Generation Options**

User-specified Procedures (recommended)  
Define user-specified request and response procedures from your WSDL schema elements. This option also generates a wrapper procedure allowing you to execute your request and response.

Default Procedures  
Generate default request and response procedures. A new procedure will be generated for each complex type. The wrapper procedure must be manually created with this option.

図9.40 WSDL ソースの選択

6. このウィザードは、クエリー可能なラップされたプロシージャーで使用されるリクエストおよび応答プロシージャーを両方生成します。次のページ **Procedure Definition** は、リクエストおよび応答構造の詳細を定義する方法を提供します。

**Request** タブで、リクエストの入力パラメーターとしたいスキーマ要素を選択およびダブルクリックします。これらは **Element Info** パネルに追加され、生成される SQL ステートメントは新しい要素を反映するために更新されます。Request および Response タブの両方にある BODY および HEADER タブに注目してください。このプロシージャー向けに選択されたサービスモードが MESSAGE に設定されると HEADER タブが有効になり、同じスキーマツリーを利用して SOAP ヘッダー変数を定義できます。

Response タブを選択し、応答プロシージャーの結果セット列を同様に作成します。

上部の Operations セレクターで選択する演算を変更し、すべての演算に対してこの処理を繰り返します。

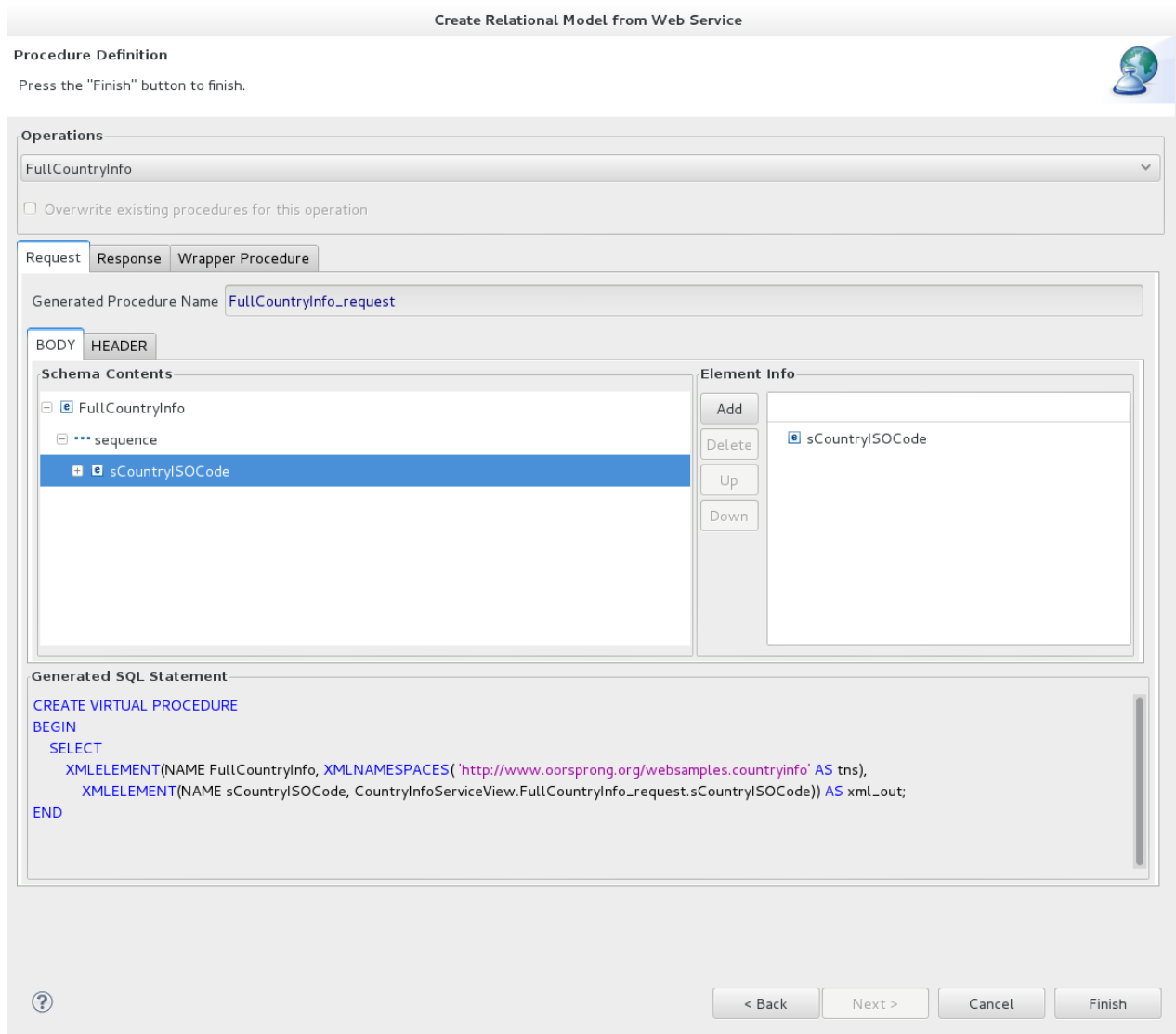


図9.41 Procedure Definition ページ

7. **Finish** をクリックします。生成後、新しいモデルはワークスペースの指定された場所に格納されます。

Model Explorer では、単一のプロシージャが含まれる単一の物理モデルが呼び出しを行った後、作成されたインポーターが表示されます。このモデルおよびプロシージャは、WSDL で宣言された単一ポートに対応します。

WSDL で演算が宣言された後、名前を付けた新しいプロシージャが含まれる単一のビューモデルも作成されます。Teiid Designer でプレビューを見ることが出来るラッパープロシージャが各演算に対して作成されます。

### 9.9.2. WSDL スキーマでの循環参照

WSDL スキーマに大変深い XML タイプ参照のセットがあったり、それらの参照が循環参照であったりする可能性があります。これは WSDL スキーマでは許可されますが、**Teiid Designer** でのスキーマの処理が困難になることがあります。このような循環参照をチェックしないと、JVM の **StackOverflow** 例外が発生し、アプリケーションが終了する可能性があります。

この問題を回避するため、深さの制限が 750 個の参照になりました。この制限を超えると、警告が表示され、それ以降の処理は終了されます。対象の参照が循環参照ではなく、単に深い参照である可能性もあるため、そのような場合は JVM プロパティ **Wsd1SchemaHandlerRecursiveDepth** により大きな値を設定

し (例: `-D WsdSchemaHandlerRecursiveDepth=800`)、深さの制限を増やすことができます。一部のシステムでは、新しい深さの制限に達する前に JVM による **StackOverflow** 例外が発生する可能性があるため、十分注意して使用するようにしてください。

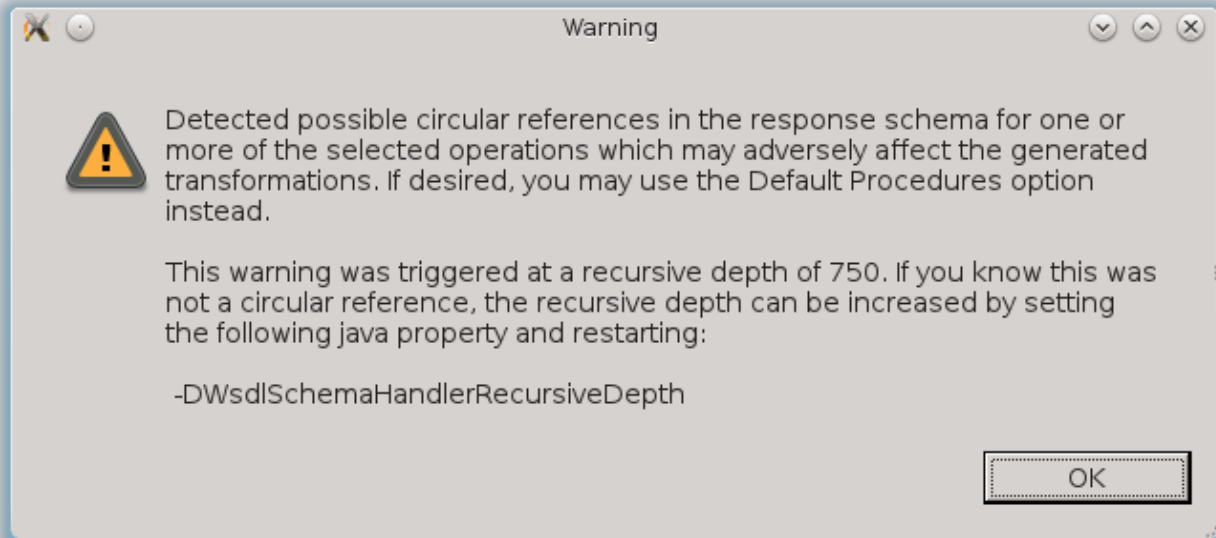


図9.42 深さの限界に達したときに表示される警告メッセージ

## 9.10. WSDL の Web サービスへのインポート

### 9.10.1. WSDL の Web サービスへのインポート

Web サービスモデルを作成するには、ワークスペースで WSDL ファイルを選択してファイルシステムから WSDL ファイルをインポートするか、URL を定義します。Teiid Designer は WSDL を解釈し、関連する XML スキーマファイルまたは依存する XML スキーマファイルを見つけます。さらに、スキーマコンポーネントの XML ビューを生成し、WSDL で定義されたインターフェースや演算を表す Web サービスモデルを作成します。


Web サービスの生成で WSDL を選択するオプションは 3 つあります。

- ✦ ワークスペースの場所
- ✦ ファイルシステムの場所
- ✦ URL

以下に、各オプションの詳細手順およびウィザードによる WSDL エラーの処理方法を説明します。

### 9.10.2. ワークスペースの場所からの WSDL のインポート

ワークスペースから WSDL ファイルを選択して Web サービスモデルを作成できます。

1. Click the **File > Import** action  in the toolbar or select a project, folder or model in the tree and click **Import...**
2. インポートオプションを **Teiid Designer > WSDL File or URL >> Web Service Model** と選択し、**Next>** をクリックします。

3. Web サービスモデルの有効な名前を入力し、**workspace...** ボタンをクリックします。ダイアログでワークスペースの WSDL ファイルを見つけ、**OK>** をクリックします。**Next>** をクリックして作業を続けます。

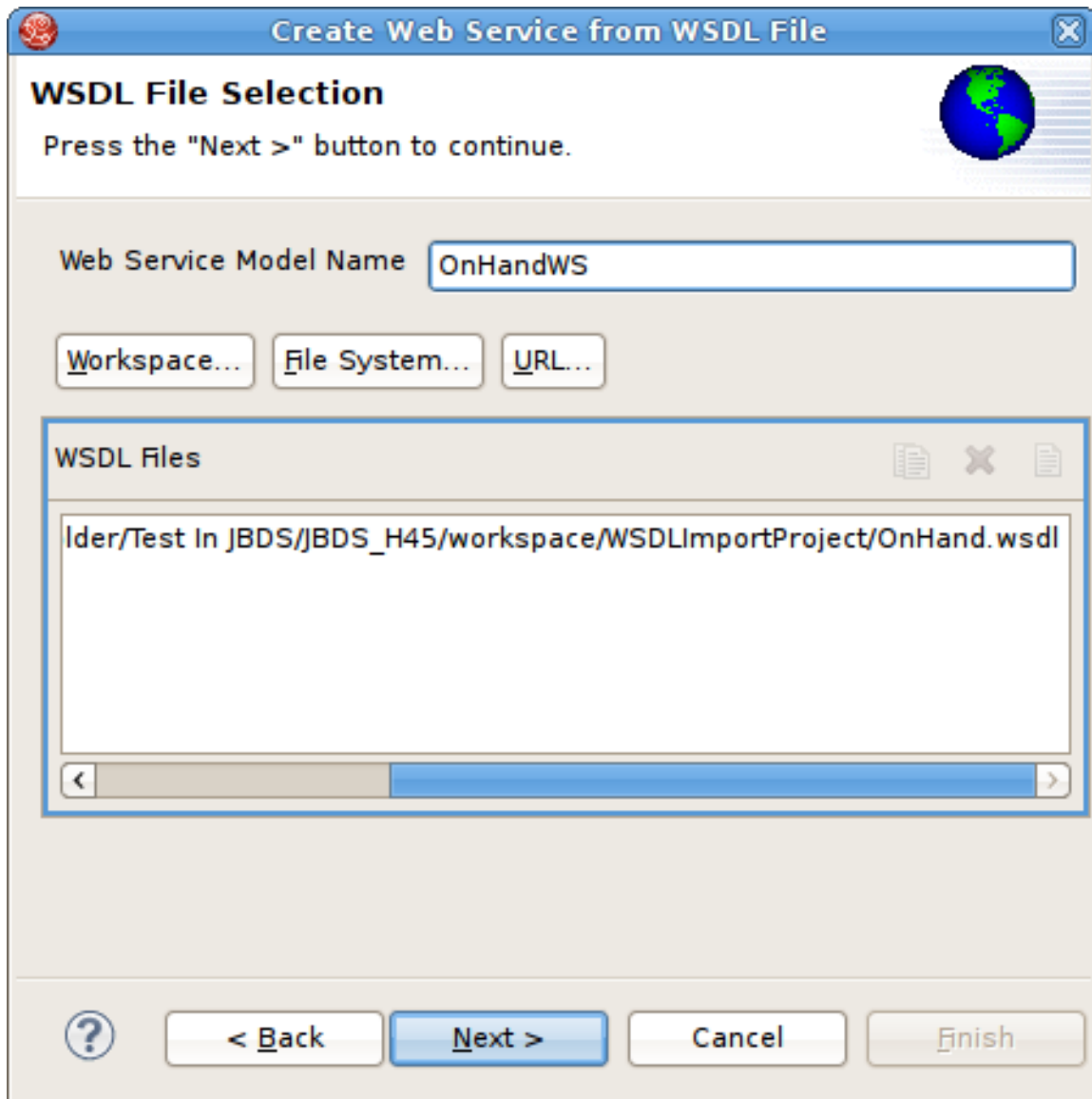


図9.43 WSDL File Selection ダイアログ

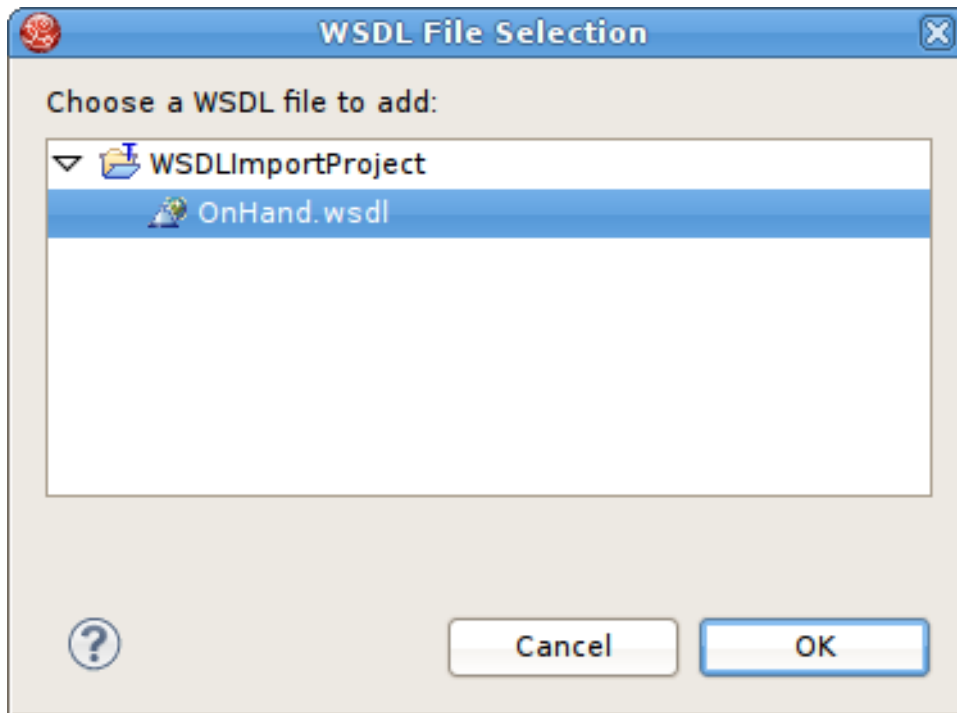


図9.44 WSDL ファイルワークスペース選択ダイアログ

**注記**

- ※ WSDL が選択または指定されないと、インポーターは空の Web サービスモデルのみを作成します。XML スキーマや XML ビューモデルは生成されません。
- ※ 参照されたファイル (WSDL またはスキーマ) は WSDL ファイルに組み込まれるか、ファイルシステムに存在する必要があります。

4. 次のページのタイトルは **Namespace Resolution** です。このページは、正常または異常な WSDL 名前空間の解決を特定します。パスの取得にワークスペースファイルセレクターが使用されるため、主な WSDL ドキュメントは基本的に常に解決されます。解決できない他の WSDL ファイルがメイン WSDL ファイルによってインポートされると問題が発生します。エラーがない場合は、**Next** をクリックして作業を続けるか、**Finish** (有効な場合) をクリックしてデフォルトのオプションを適用して終了します。

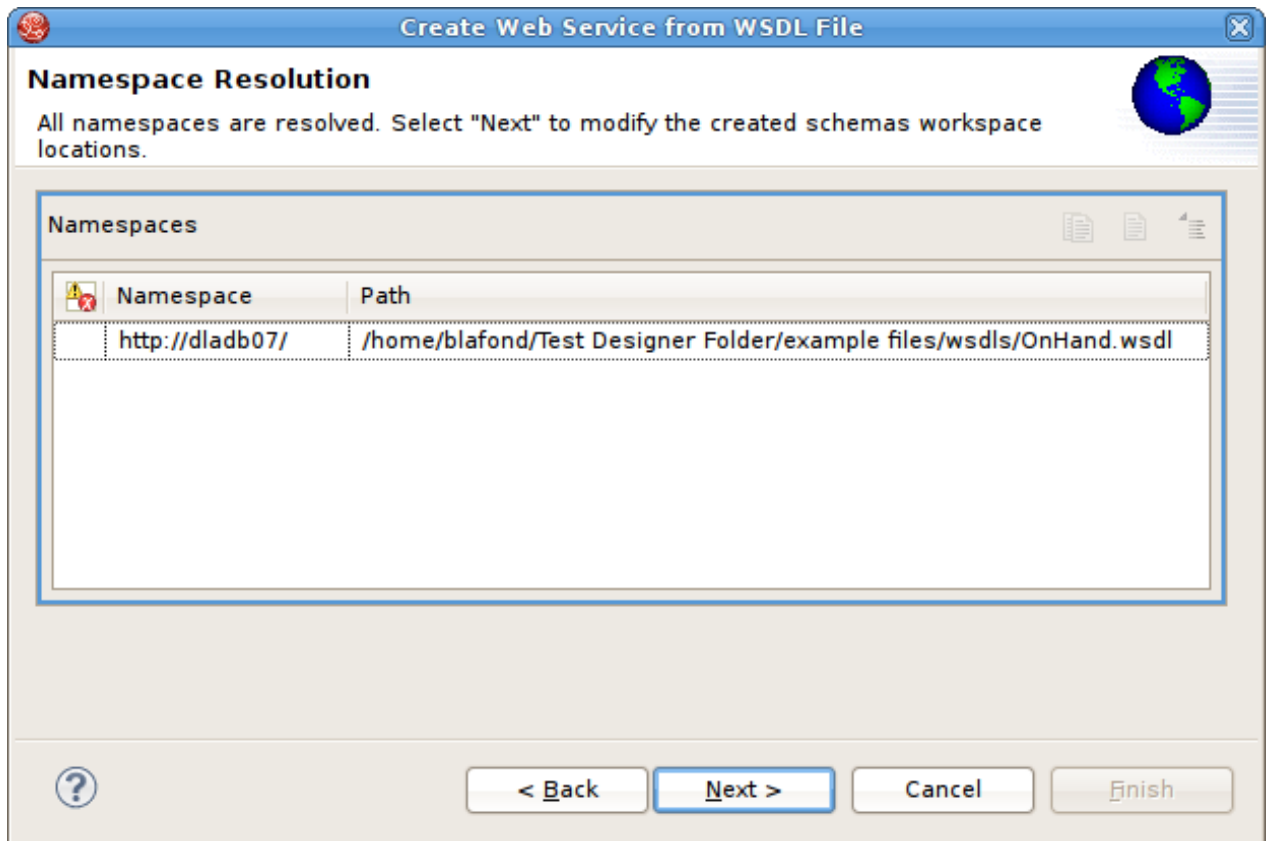


図9.45 Namespace Resolution ダイアログ

5. 次のページは **WSDL Operations Selection** です。ここでは、次のダイアログでさまざまな演算やインターフェースを選択または選択解除し、Web サービスモデルのコンテンツをカスタマイズできます。

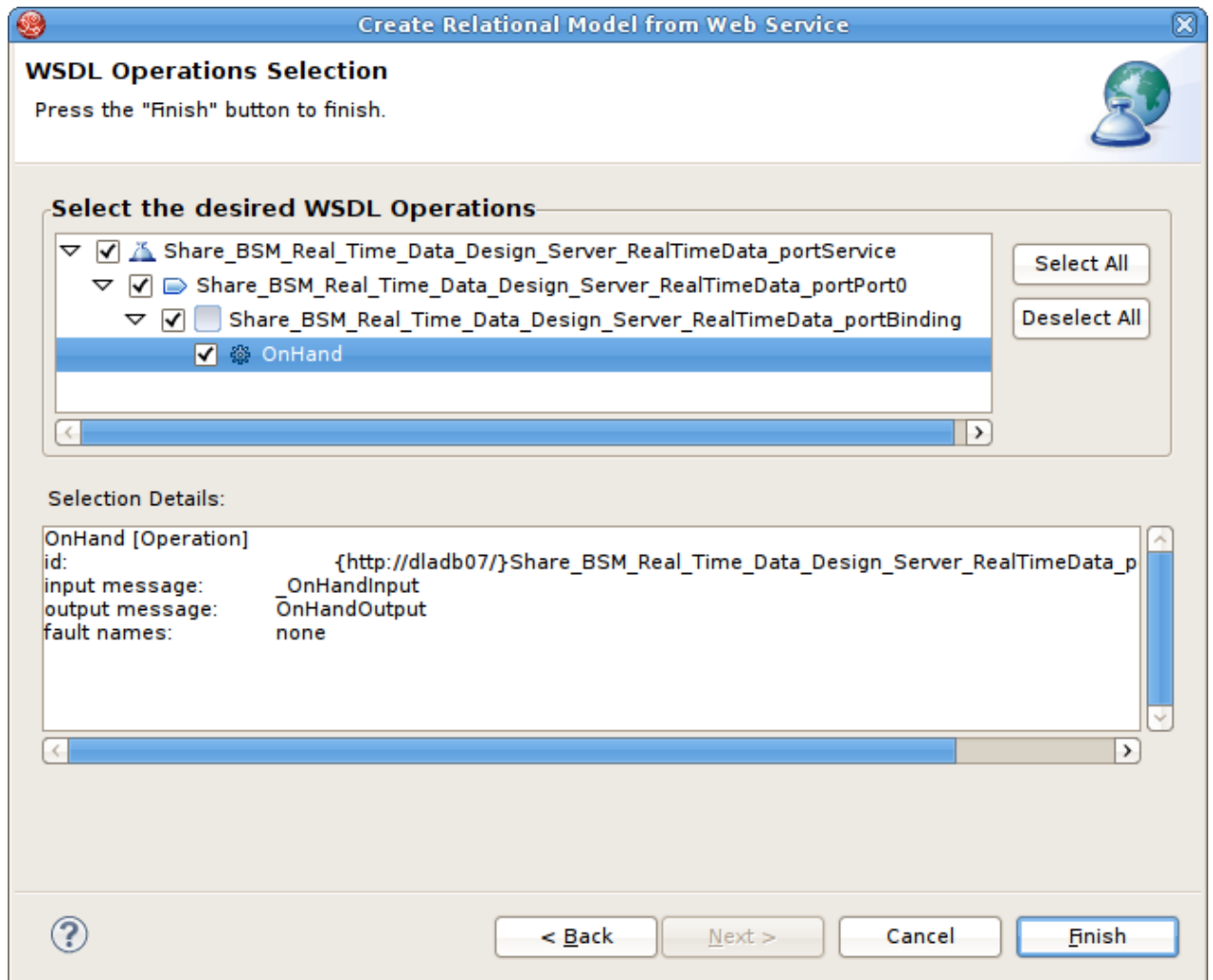


図9.46 Namespace Resolution ダイアログ

6. 次のページのタイトルは **Schema Workspace Location Selection** です。このページでは、WSDL によってインポートされるすべてのスキーマ (およびスキーマ内で参照される依存スキーマ) と WSDL に埋め込まれたスキーマをリストし、これらが解決可能であるかどうかを示します。解決されたスキーマはすべて別のファイルに作成され、ワークスペースに追加されます。エディターパネルでは、新しいスキーマファイルのデフォルトのファイル名を変更できます。

エラーがない場合は、**Next** をクリックして作業を続けるか、**Finish** をクリックしてデフォルトのオプションを適用して終了します。

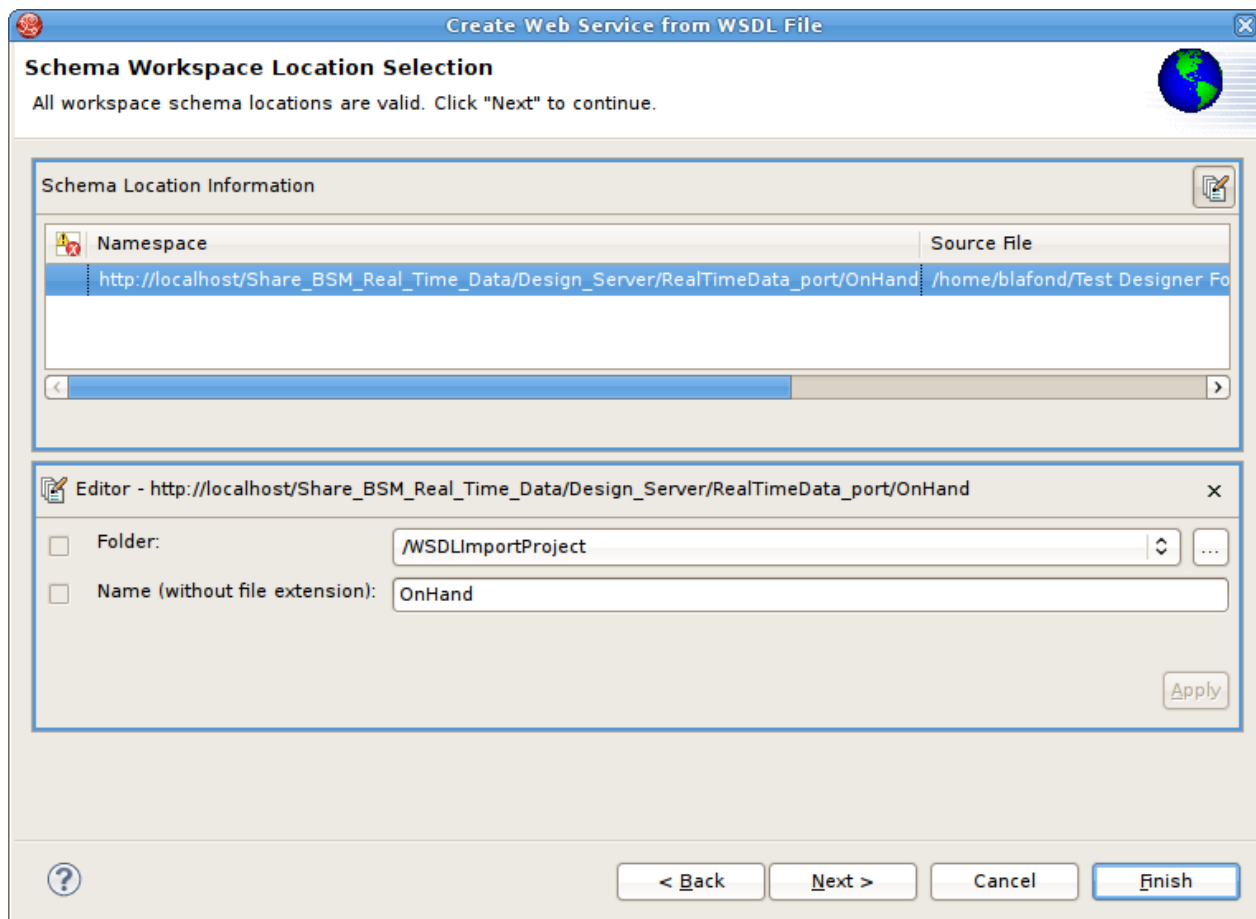


図9.47 Namespace Resolution ダイアログ

- 最後のページのタイトルは **XML Model Generation** で、「Generate virtual XML document model」を選択した場合に XML ビューモデルの名前を変更できます。希望の名前を入力するか、提供されるデフォルトの名前を使用します。**Finish** をクリックして完了します。



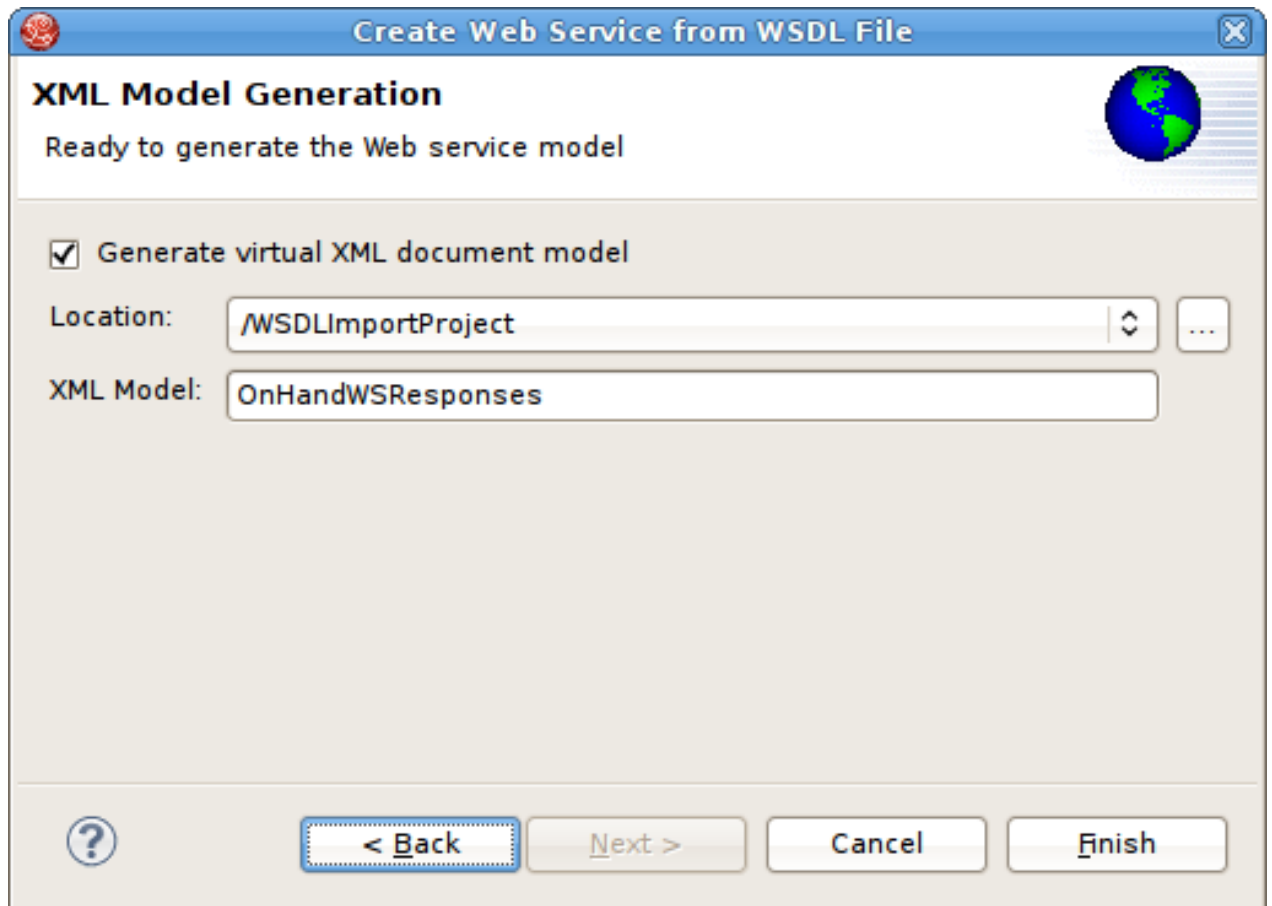


図9.48 Namespace Resolution ダイアログ

WSDL から正常に Web サービスを生成するには、WSDL にエラーがないようにする必要があります。WSDL の検証は上記のステップ 3 で実行されます。エラーが存在する場合は、以下のようなエラーの概要を示すダイアログが表示され、WSDL の問題が修正されるか、有効な WSDL ファイルが再インポートおよび選択されるまでウィザードを終了することはできません。

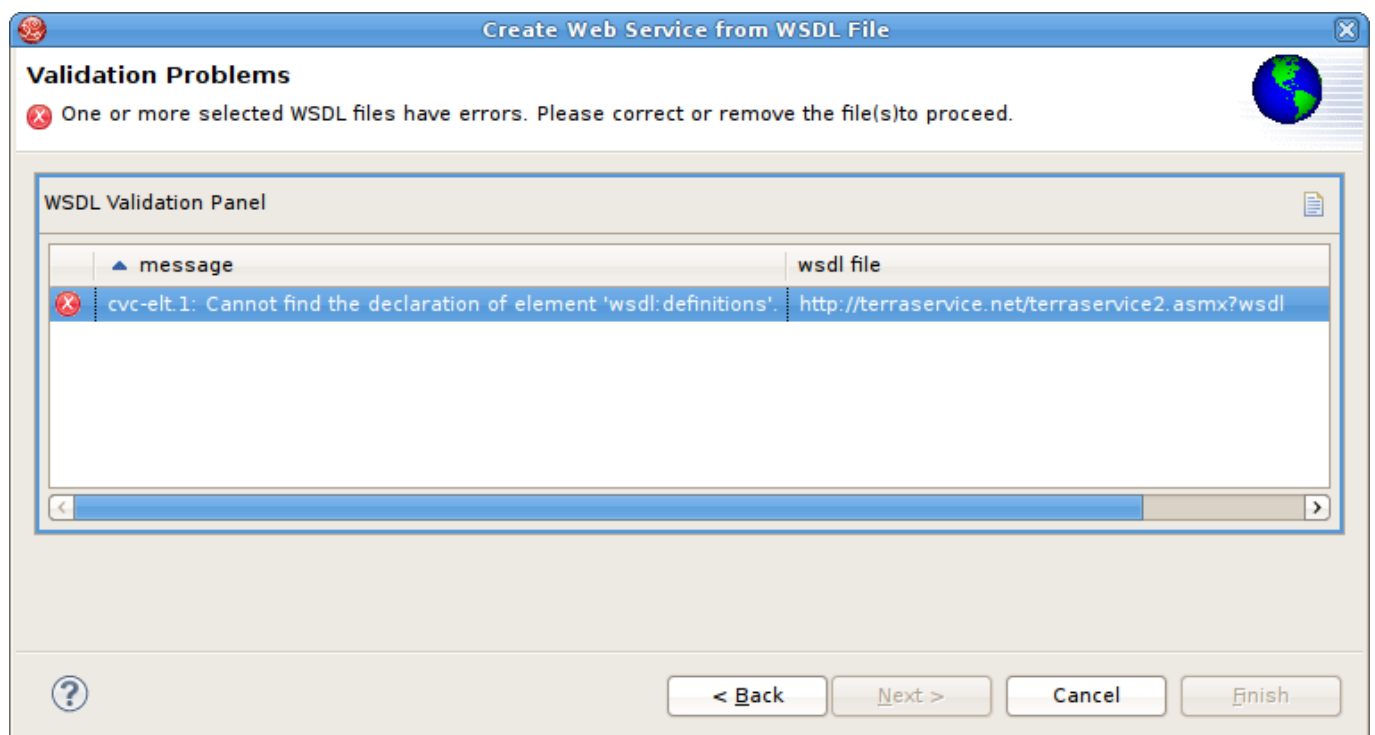



図9.49 WSDL Validation Problems ダイアログ

### 9.10.3. ファイルシステムの場所からの WSDL のインポート

ローカルファイルシステムから WSDL ファイルを選択して Web サービスモデルを作成できます。

1. Click the **File** > **Import** action  in the toolbar or select a project, folder or model in the tree and click **Import...**
2. インポートオプションを **Teiid Designer** > **WSDL File or URL** >> **Web Service Model** と選択し、**Next>** をクリックします。
3. Web サービスモデルの有効な名前を入力し、**File System...** ボタンをクリックします。ダイアログでファイルシステムの WSDL ファイルを見つけ、**OK>** をクリックします。

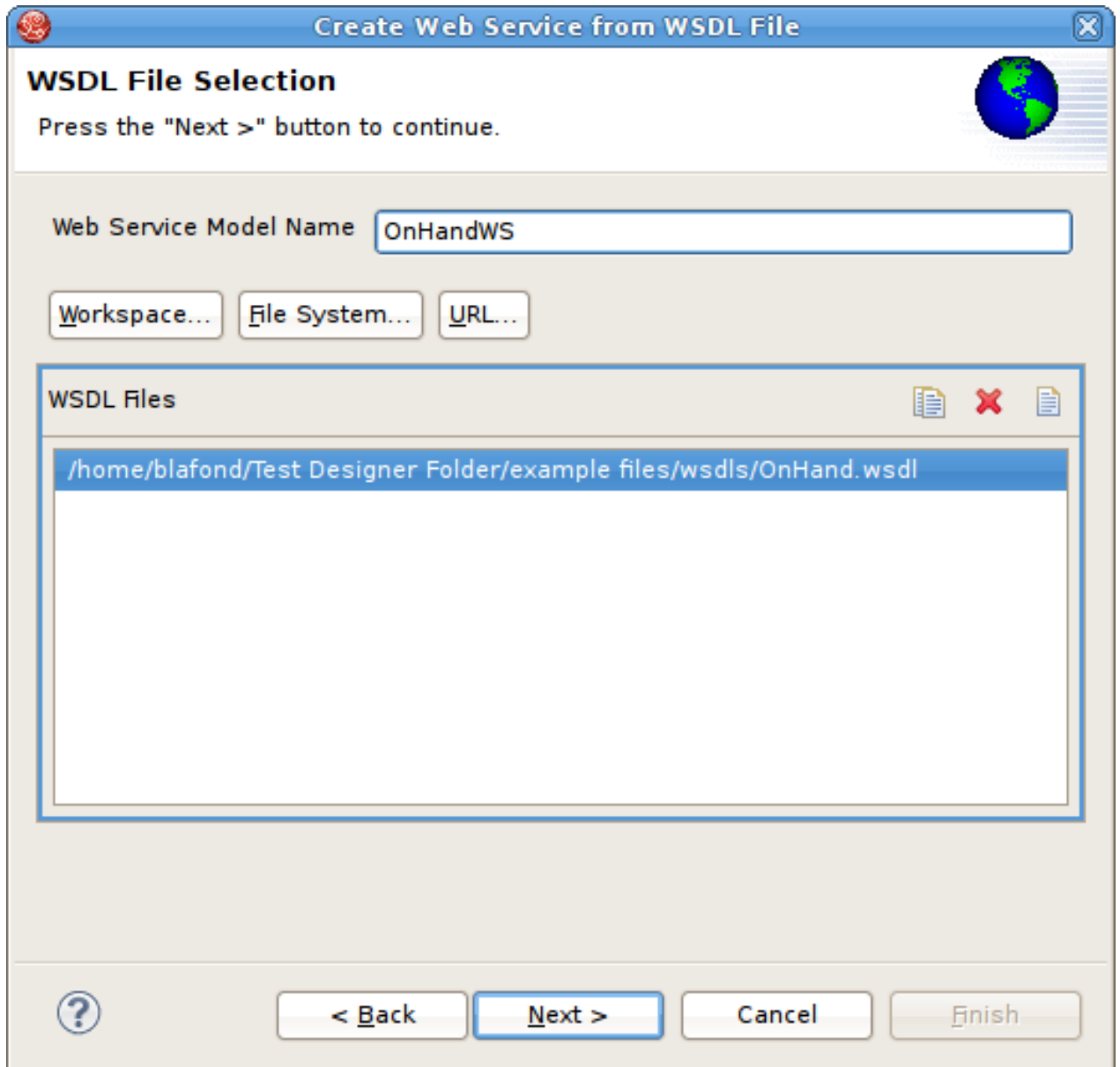


図9.50 WSDL File Selection ダイアログ

## 注記

- ※ WSDL が選択または指定されないと、インポーターは空の Web サービスモデルのみを作成します。XML スキーマや XML ビューモデルは生成されません。
- ※ 参照されたファイル (WSDL またはスキーマ) は WSDL ファイルに組み込まれるか、ファイルシステムに存在する必要があります。

4. 次のページのタイトルは **Namespace Resolution** です。このページは、正常または異常な WSDL 名前空間の解決を特定します。パスの取得にワークスペースファイルセクターが使用されるため、主な WSDL ドキュメントは基本的に常に解決されます。解決できない他の WSDL ファイルがメイン WSDL ファイルによってインポートされると問題が発生します。エラーがない場合は、**Next** をクリックして作業を続けるか、**Finish** (有効な場合) をクリックしてデフォルトのオプションを適用して終了します。

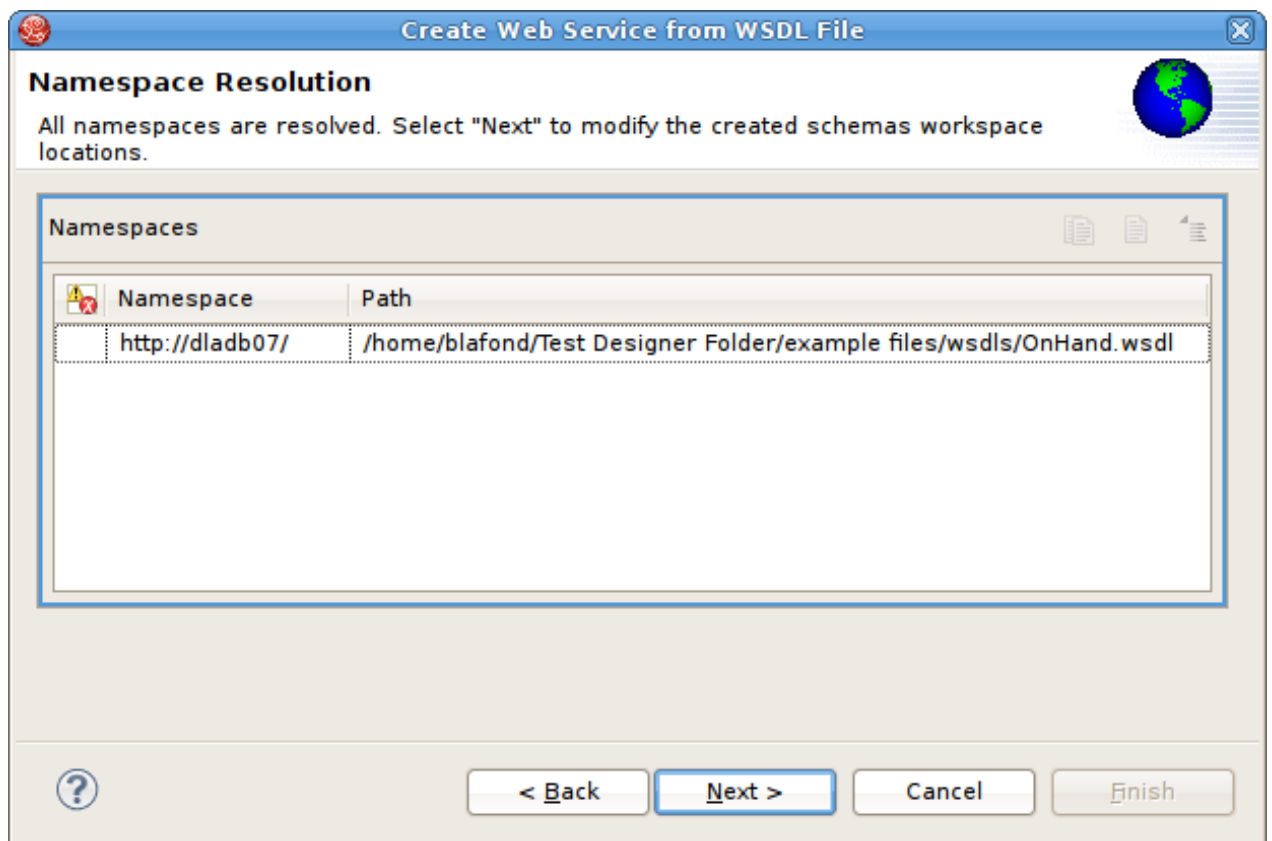


図9.51 Namespace Resolution ダイアログ

5. 次のページは **WSDL Operations Selection** です。ここでは、次のダイアログでさまざまな演算やインターフェースを選択または選択解除し、Web サービスモデルのコンテンツをカスタマイズできます。

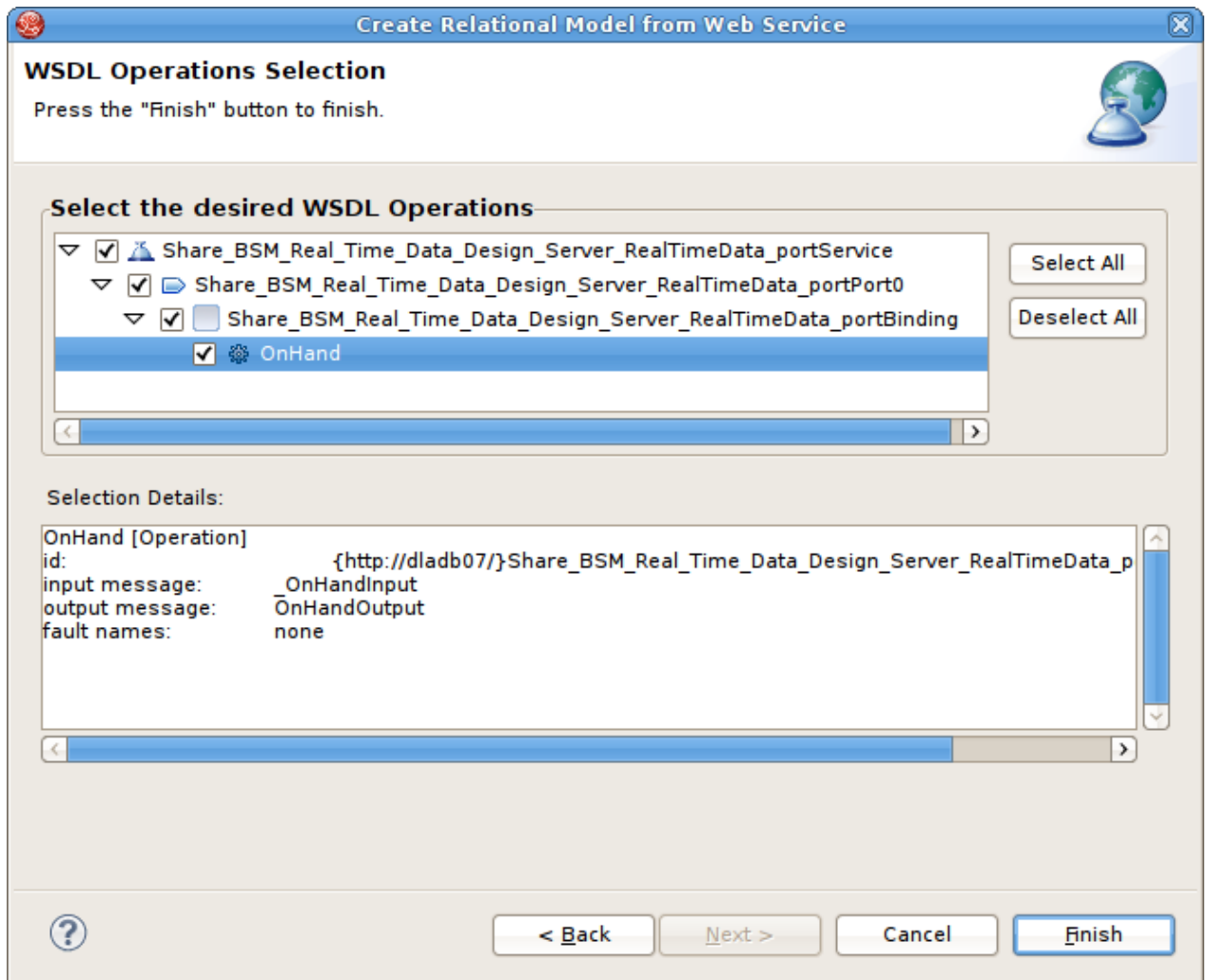


図9.52 Namespace Resolution ダイアログ

6. 次のページのタイトルは **Schema Workspace Location Selection** です。このページでは、WSDL によってインポートされるすべてのスキーマ (およびスキーマ内で参照される依存スキーマ) と WSDL に埋め込まれたスキーマをリストし、これらが解決可能であるかどうかを示します。解決されたスキーマはすべて別のファイルに作成され、ワークスペースに追加されます。エディターパネルでは、新しいスキーマファイルのデフォルトのファイル名を変更できます。

エラーがない場合は、**Next** をクリックして作業を続けるか、**Finish** をクリックしてデフォルトのオプションを適用して終了します。

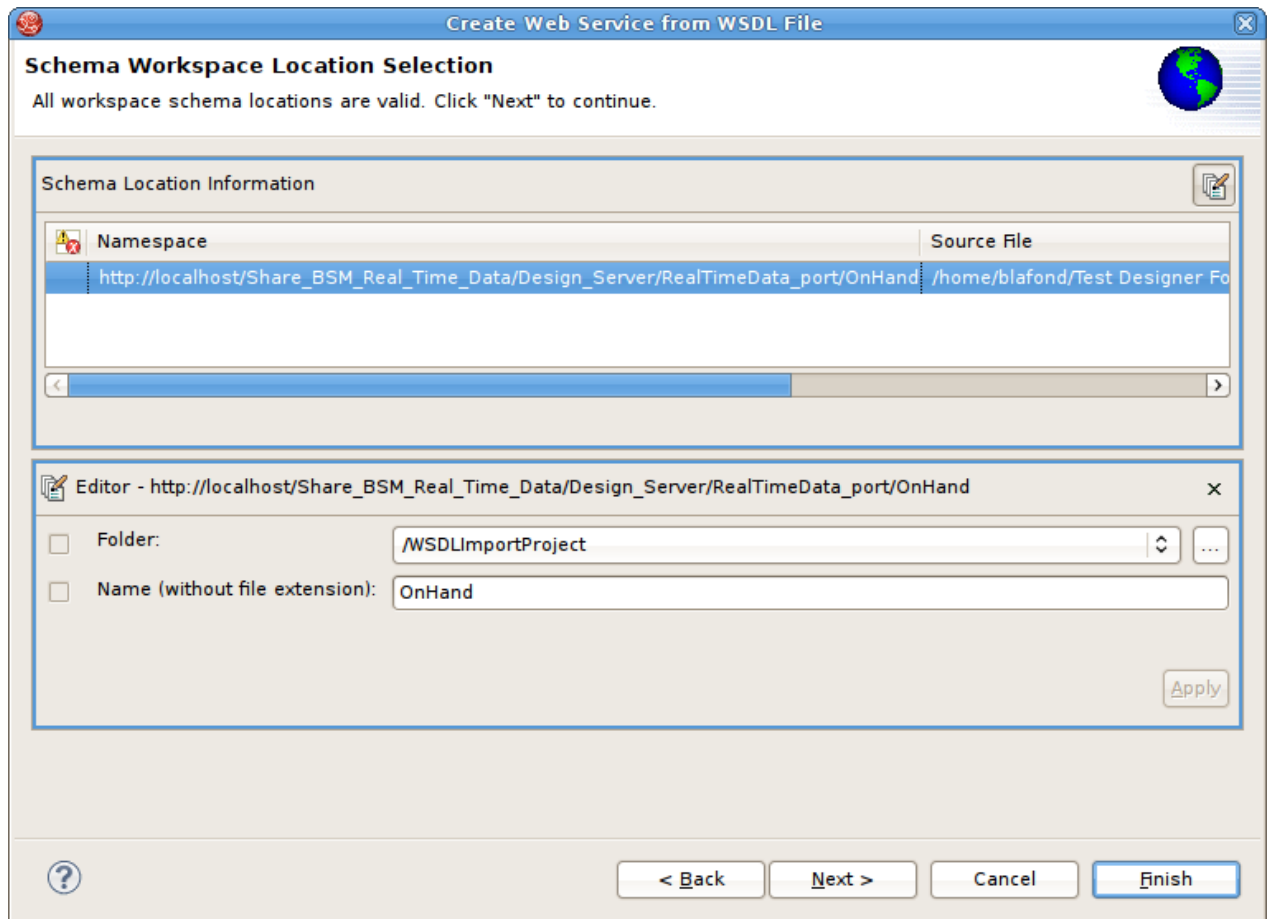


図9.53 Namespace Resolution ダイアログ

- 最後のページのタイトルは **XML Model Generation** で、「Generate virtual XML document model」を選択した場合に XML ビューモデルの名前を変更できます。希望の名前を入力するか、提供されるデフォルトの名前を使用します。**Finish** をクリックして完了します。

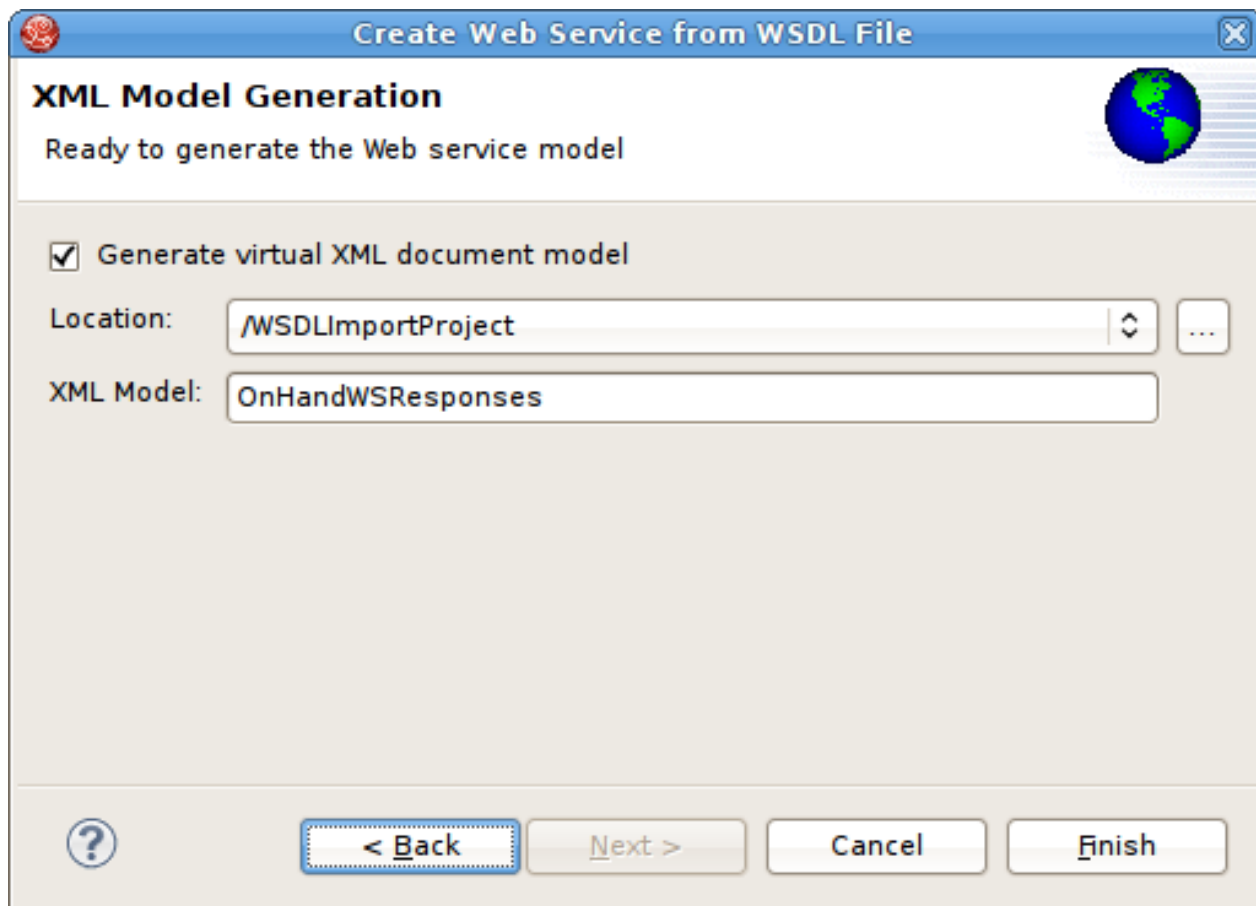


図9.54 Namespace Resolution ダイアログ

WSDL から正常に Web サービスを生成するには、WSDL にエラーがないようにする必要があります。WSDL の検証は上記のステップ 3 で実行されます。エラーが存在する場合は、以下のようなエラーの概要を示すダイアログが表示され、WSDL の問題が修正されるか、有効な WSDL ファイルが再インポートおよび選択されるまでウィザードを終了することはできません。

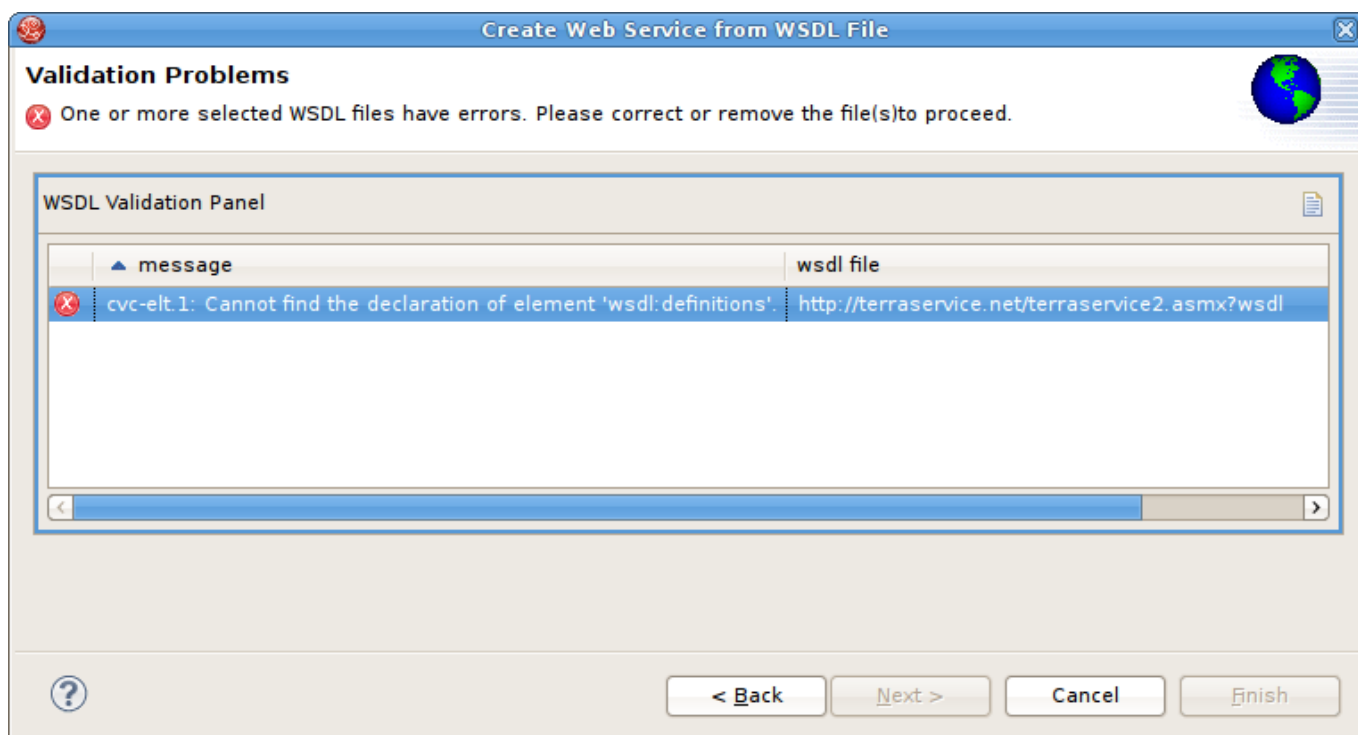



図9.55 WSDL Validation Problems ダイアログ

### 9.10.4. URL からの WSDL のインポート

ワークスペースから WSDL ファイルを選択して Web サービスモデルを作成できます。

1. Click the **File** > **Import** action  in the toolbar or select a project, folder or model in the tree and click **Import...**
2. インポートオプションを **Teiid Designer** > **WSDL File or URL** >> **Web Service Model** と選択し、**Next>** をクリックします。
3. Web サービスモデルの有効な名前を入力し、**URL...** ボタンをクリックします。
  - a. 有効な WSDL URL を入力します。URL を検証できない場合はエラーが表示され、**OK>** ボタンが無効化されます。
  - b. WSDL が基本の HTTP 認証によって保護されている場合は、このオプションを選択し、適切なユーザー名とパスワードを入力する必要があります。
  - c. **OK>** をクリックして次に進みます。

**Next>** をクリックして次に進みます。

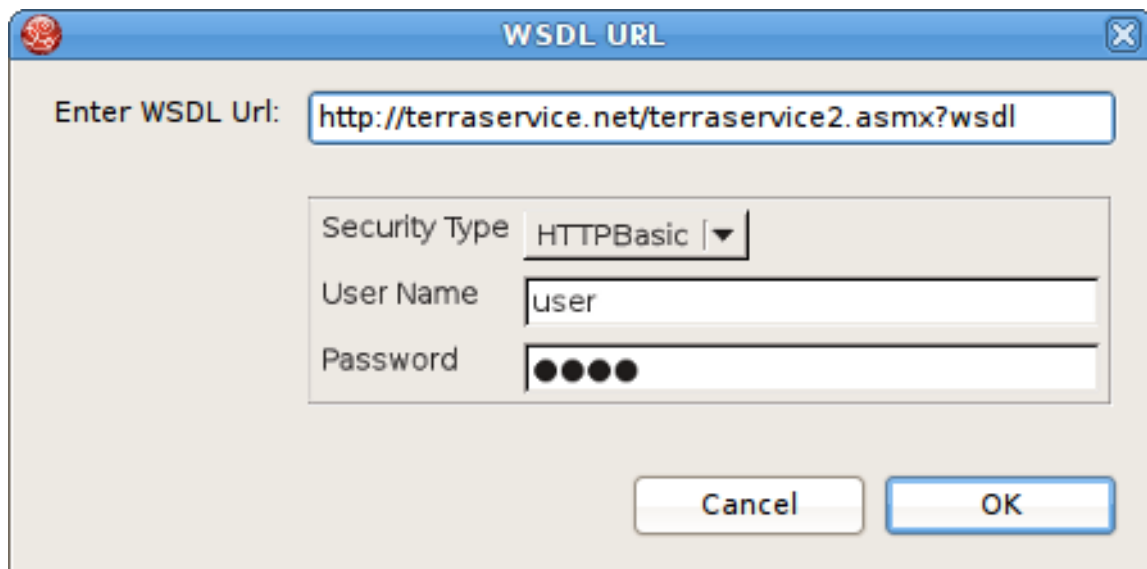


図9.56 WSDL URL ダイアログ



#### 注記

- ※ WSDL が選択または指定されないと、インポーターは空の Web サービスモデルのみを作成します。XML スキーマや XML ビューモデルは生成されません。
- ※ 参照されたファイル (WSDL またはスキーマ) は WSDL ファイルに組み込まれるか、ファイルシステムに存在する必要があります。

4. 次のページのタイトルは **Namespace Resolution** です。このページは、正常または異常な WSDL 名前空間の解決を特定します。パスの取得にワークスペースファイルセレクターが使用されるため、主な WSDL ドキュメントは基本的に常に解決されます。解決できない他の WSDL ファイルがメイン WSDL ファイルによってインポートされると問題が発生します。エラーがない場合は、**Next** をクリックして作業を続けるか、**Finish** (有効な場合) をクリックしてデフォルトのオプションを適用して終了します。

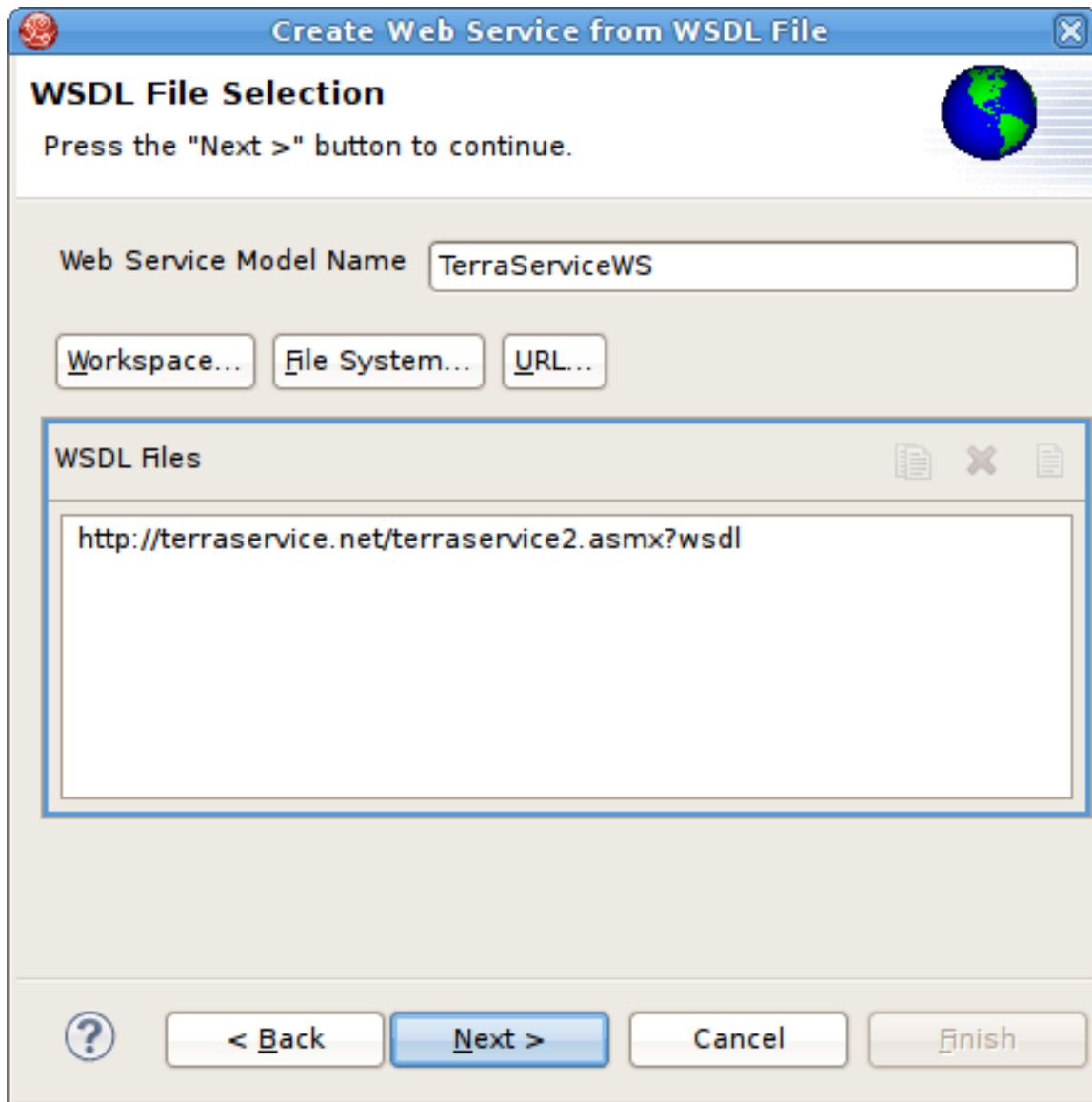


図9.57 Namespace Resolution ダイアログ

5. 次のページは **WSDL Operations Selection** です。ここでは、次のダイアログでさまざまな演算やインターフェースを選択または選択解除し、Web サービスモデルのコンテンツをカスタマイズできます。



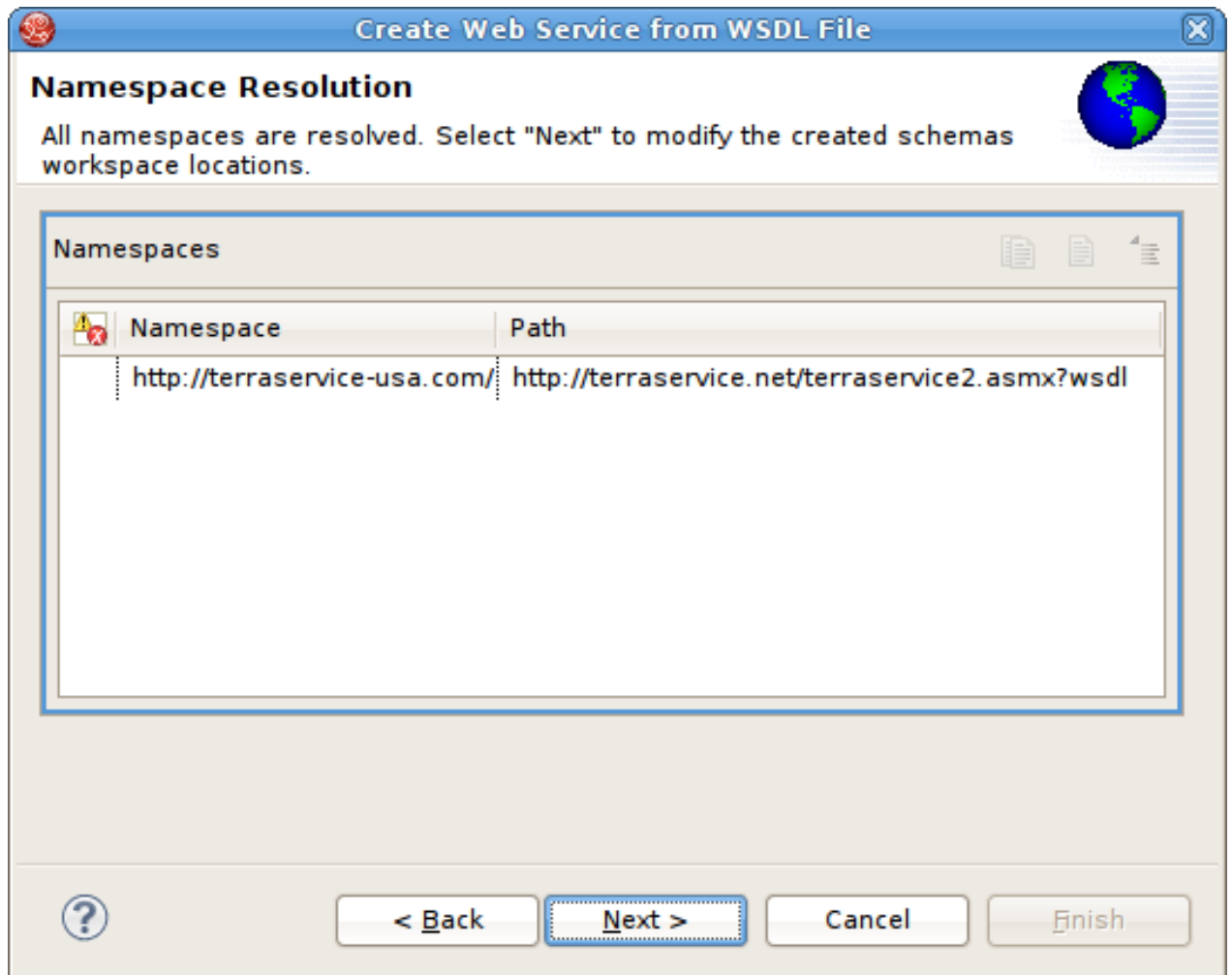


図9.58 Namespace Resolution ダイアログ

- 次のページのタイトルは **Schema Workspace Location Selection** です。このページでは、WSDL によってインポートされるすべてのスキーマ (およびスキーマ内で参照される依存スキーマ) と WSDL に埋め込まれたスキーマをリストし、これらが解決可能であるかどうかを示します。解決されたスキーマはすべて別のファイルに作成され、ワークスペースに追加されます。エディターパネルでは、新しいスキーマファイルのデフォルトのファイル名を変更できます。

エラーがない場合は、**Next** をクリックして作業を続けるか、**Finish** をクリックしてデフォルトのオプションを適用して終了します。

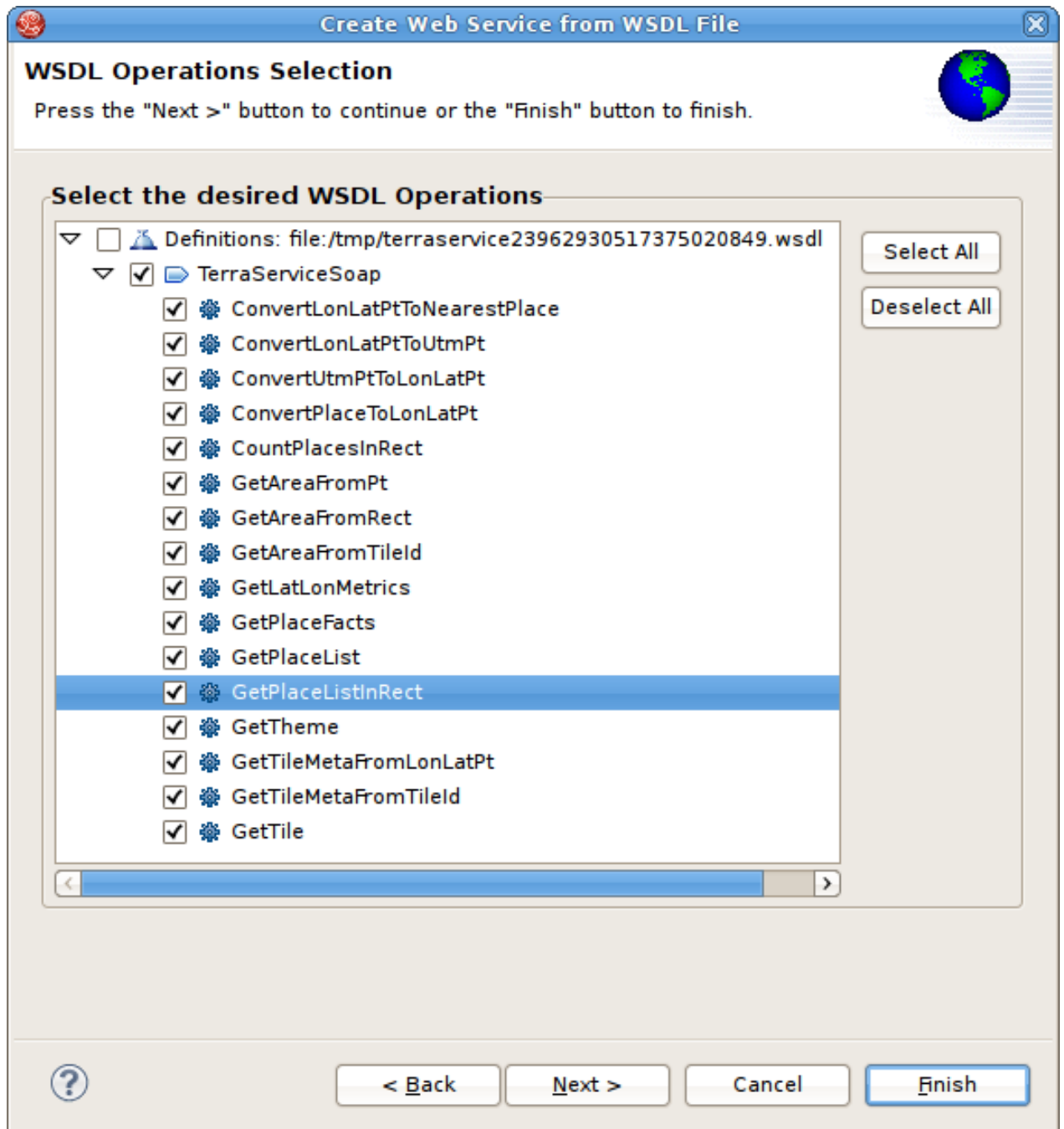


図9.59 Namespace Resolution ダイアログ

- 最後のページのタイトルは **XML Model Generation** で、「Generate virtual XML document model」を選択した場合に XML ビューモデルの名前を変更できます。希望の名前を入力するか、提供されるデフォルトの名前を使用します。 **Finish** をクリックして完了します。

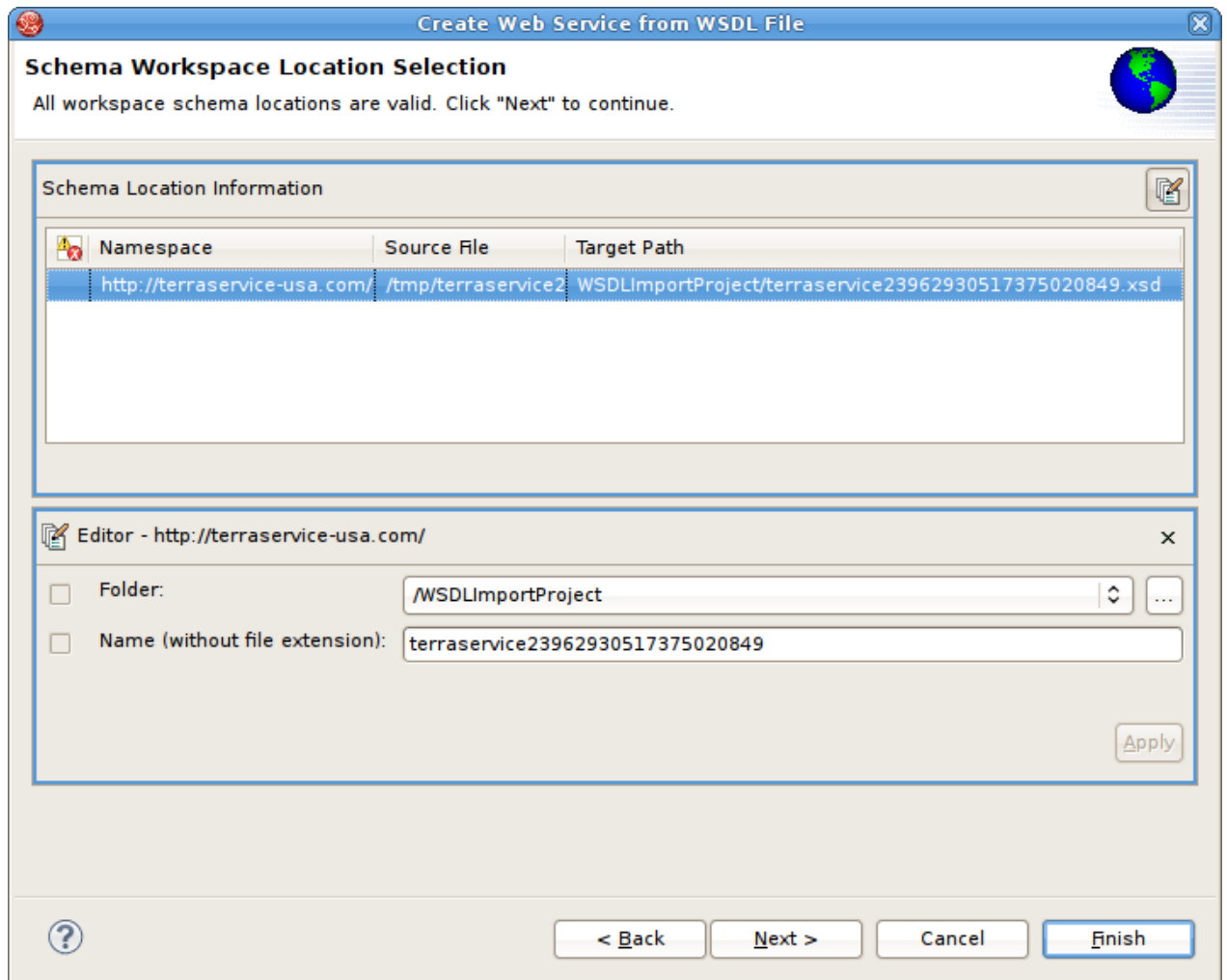


図9.60 Namespace Resolution ダイアログ

WSDL から正常に Web サービスを生成するには、WSDL にエラーがないようにする必要があります。WSDL の検証は上記のステップ 3 で実行されます。エラーが存在する場合は、以下のようなエラーの概要を示すダイアログが表示され、WSDL の問題が修正されるか、有効な WSDL ファイルが再インポートおよび選択されるまでウィザードを終了することはできません。

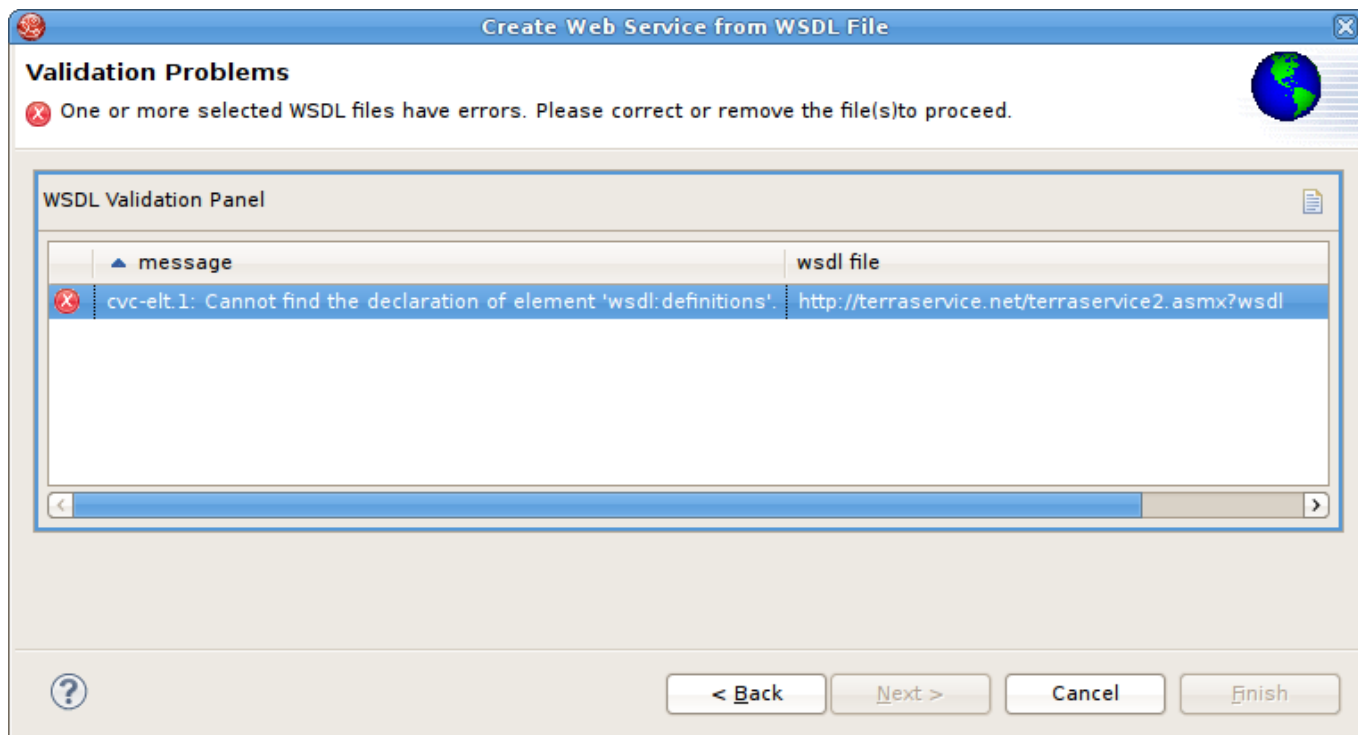


図9.61 WSDL Validation Problems ダイアログ

## 9.11. REST サービスからのデータのインポート

### 手順9.2 REST サービスからのデータのインポート

1. **Model Explorer** に移動します。
2. ツールバーの **File - Import** アクションを選択します。



#### 注記

この代わりに、**tree** でプロジェクト、フォルダー、またはモデルを選択し、**Import...** を選択することもできます。

3. インポートオプション **Web Service Source >> Source and View Model (REST)** を選択し、**Next** をクリックします。
4. ドロップダウンメニューから既存または以前の接続プロファイルを選択します。  
この代わりに、**New...** ボタンを押して **New Connection Profile** ダイアログを起動するか、選択前に **Edit...** を押して既存の接続プロファイルを変更することもできます。
5. 新しい接続プロファイルを作成する場合は、**Web Services Data Source (REST)** を選択します。
6. 新しい REST 接続プロファイルの名前を付け、**Next** をクリックします。
7. **Connection URL** テキストフィールドで REST URL を設定し、該当する場合はセキュリティタイプとクレデンシャルを設定します。**Next** をクリックしてプロパティの概要を表示するか、**Finish** をクリックして終了します。

Request Header パラメーターを追加するオプションもあります。インポーターは、アプリケーションや xml の Accept ヘッダー値および Content-Type ヘッダー値を仮定します。これらのデフォルト

は Optional Request Header Properties セクションで上書きできます。また、サービスに必要なその他のヘッダープロパティを追加することもできます。**Test Connection** ボタンをクリックし、接続プロパティを検証します。

- 新しい接続プロファイルを選択または作成した後、接続プロファイルからの REST XML の結果が REST Web Service Response Data File パネルに表示されます。処理するデータファイルをチェックして、Next をクリックします。

- Source Model Definition セクションに一意のソースモデル名を入力します。

または、Browse ボタンを使用して既存のソースモデルを選択します。



### 注記

Model Status は、モデル名の有効性、モデルが存在するかどうか、およびモデルに invokeHttp() プロシージャがすでに含まれているかどうかを示します。この場合、ソースモデルもプロシージャも生成されません。

- View Model Definition セクションに一意のビューモデル名を入力します。

または、Browse ボタンを使用して既存のビューモデルを選択します。



### 注記

Model Status は、モデル名の有効性とモデルが存在するかどうかを示します。

- View Model Definition セクションに新規のビュープロシージャ名を入力して、Next をクリックします。

- Jboss Data Source Information セクションに JNDI 名を入力して、Next をクリックします。



### 注記

このインポーターの主な目的は、ユーザー定義のデータファイルをクエリーするために必要なトランスフォーメーションが含まれるビュープロシージャを作成することです。パネルには、結果のコンテンツの XML ツリービューと、中央の Column Information パネルに表示される列エントリの作成に使用できるアクションやボタンが含まれます。ルートパスは、結果のドキュメントの xpath 解析に使用されます。インポーターによってルートパスが設定されます。必要な場合は XML 要素を選択し、「Set as root path」を右クリックするとルートパスを変更できます。次に、ツリーでクエリーに含める列を選択し、Add ボタンを選択します。また、カスタム列を変更または作成するには、ADD、DELETE、EDIT、UP、DOWN を使用して SQL の列情報を管理します。

- Finish をクリックしてモデルを生成し、ウィザードを終了します。

## 9.12. LDAP (Lightweight Directory Access Protocol)

LDAP (Lightweight Directory Access Protocol) は TCP/IP 上で分散されたディレクトリー情報へアクセスするプロトコルです。ディレクトリーには、ツリー形式で階層的に構成された関連データの集まりが含まれています。ツリーの各ノードはディレクトリーエントリーで、各エントリーは属性と値のペアのセットで構成さ

れます。各ディレクトリーエントリーは識別名 (DN) と呼ばれる一意の識別子を持ちます。DN は、エントリー自体の属性から構築された RDN (相対識別名) とその後続く親エントリーの DN で構成されます。

## 9.13. LDAP サーバーからのインポート

Teiid Designer では、LDAP サーバーのモデリングは以下のようになります。

- ※ LDAP サブツリーはリレーショナルデータベースのテーブルのように表されます。
- ※ サブツリーの各ノードは、テーブルの行として表されます。
- ※ ノードの各属性は、テーブルの列として表すことができます。
- ※ RDN (または DN) を使用してプライマリーキーを表すことができます。

LDAP メタデータはリレーショナルメタモデルを使用してモデル化されます。リレーショナルモデルの各テーブルはディレクトリーエントリーを表し、テーブルの各行はディレクトリーエントリーの子エントリーを表します。テーブルの各列は、存在する可能性がある子エントリーの属性を表します。通常、各テーブルおよび列はプロパティ「Name In Source」の LDAP 固有の情報を定義します。これにより、コネクターは LDAP 内などのデータソース内で属性またはベース DN 名を識別できます。テーブルおよび列の実際の名前はソースの名前とは異なることがあり、モデルとクエリーでより記述的なラベル付けが可能になります。

LDAP データからリレーショナルモデルを作成する方法は、以下のようになります。

1. Model Explorer でツールバーの File > Import アクションをクリックするか、ツリーのプロジェクト、フォルダー、またはモデルを選択し、Import を選択します。
2. インポートオプション Teiid Designer > LDAP Service >> Source Mode を選択し、Next をクリックします。
3. ドロップダウンメニューから既存または以前の LDAP 接続プロファイルを選択するか、New... ボタンをクリックして New Connection Profile ダイアログを開始します (JBDS Data Tools ドキュメントを参照)。または、Edit... をクリックして選択前に既存の接続プロファイルを変更します。



### 注記

新規接続で必要となる情報は、以下の通りです。

- ※ 接続のユーザー名とパスワード - LDAP ツリーを閲覧する管理者アカウント (例: cn=Manager,dc=birds-of-prey,dc=org)
- ※ 接続 URL (例: ldap://falcon:389)
- ※ プリンシパルの識別名 (DN) 接尾辞 - LDAP ツリーのルート DN
- ※ LDAP 接続ファクトリーの実装クラス (例: com.sun.jndi.ldap.LdapCtxFactory)

接続プロファイルを選択すると、LDAP サービス URL と DN 接尾辞 フィールドが入力されます。ウィザードページの残りの要件は、インポートされたテーブルの宛先として適切なモデルファイルを選択することです。既存のモデルを選択した場合、ウィザードは新しいテーブルをモデルの現在のコンテンツにマージします。

4. 接続プロファイルを選択したら Next をクリックします。
5. Select LDAP Entries to be modeled as tables ページで、ソースモデルのテーブルとして作成される LDAP エントリーをツリーから選択します (ツリーの各チェックボックスにチェックマークを入れ、エントリーを選択します)。エントリーをハイライトすると、以下の属性が表示されます。
  - ※ Table Name (テーブル名) - テーブルのラベルで、より読みやすい値に変更できます。

- ※ Table Source Name (テーブルソース名) - 完全修飾エントリー名。ウィザードでは編集できず、後で作成されるソースモデルでも変更しないようにしてください。
  - ※ Table Source Name Suffix (テーブルソース名接尾辞) - テーブルの検索基準の範囲をさらに限定するため追加の接尾辞を追加できます。接尾辞の形式は「?search\_scope?objectClass\_name」で、search\_scope は OBJECT\_SCOPE (最初と 1 つのエントリーのみが返されます)、ONELEVEL\_SCOPE (選択されたエントリーの直下にあるエントリーのみが返されます)、または SUBTREE\_SCOPE (選択されたエントリー下にあるすべてのエントリーが再帰的に返されます) の 1 つになります。objectClass\_name は LDAP ツリーにある objectClass の特定タイプの名前です (例: inetOrgPerson エントリーのみを返します)。基準は両方とも任意 (「?」は必須) であるため、「??inetOrgPerson」のような接尾辞が可能です。
6. Next をクリックします。
  7. Select the LDAP Attributes to be modeled as columns ページは以前選択された LDAP エントリーとそれらの子エントリーの属性を表示します。選択された属性は、関連するソースモデルテーブルで列として作成することを目的としています。ツリーのチェックボックスにチェックマークを入れて属性を選択します。エントリーをハイライトすると、以下の属性が表示されます。
    - ※ Column Name (列名) - 列のラベルで、より読みやすい値に変更できます。
    - ※ Column Source Name (列ソース名) - 実際の LDAP 属性名。ウィザードでは編集できず、後で作成されるソースモデルでも変更しないようにしてください。
    - ※ Column Distinct Value Count (列の個別値数) - LDAP サービスの特定の属性に割り当てられた個別値数。この値は、ソースモデルを使用してクエリーを最適化するのに便利です。ウィザードでは編集できず、後で作成されるソースモデルでも変更しないようにしてください。
    - ※ Column Null Value Count (列の Null 値数) - LDAP サービスで特定の属性に値が割り当てされていないエントリーの数。この値は、ソースモデルを使用してクエリーを最適化するのに便利です。ウィザードでは編集できず、後で作成されるソースモデルでも変更しないようにしてください。
    - ※ Column Length (列の長さ) - LDAP サービスの属性に割り当てられた既存値の最大長。この値は、列の最大長として割り当てられます。ウィザードでは編集できませんが、必要な場合は後でソースモデルでの編集が可能です。
  8. Finish をクリックします。
  9. ウィザードが完了したら、新しいソースモデルが作成されます。

## 9.14. LDAP コネクタの更新機能

LDAP 接続は、更新機能も提供しますが、読み取り専用アクセスの必要性以外に追加のモデル化要件が適用されます。追加の要件は次のとおりです。

- ※ Update テーブルプロパティのサポート - 更新を有効にするには、各ソースモデルテーブルでこのプロパティを true に設定する必要があります。
- ※ 更新可能な column プロパティ - 更新を有効にするには、ソースモデルテーブルの各列でこのプロパティを true に設定する必要があります。
- ※ さらに、以下の列が追加で必要になります。
  - DN - すべての更新タイプ (INSERT、UPDATE、および DELETE) で、ソースの名前を dn に設定し、識別名を列としてモデル化する必要があります。UPDATE および DELETE では DN を criteria 句で指定する必要があります。INSERT では DN が設定される列値の 1 つである必要があります。
  - objectClass - INSERT では、ソースの名前を objectClass に設定し、オブジェクトクラスを列として

モデル化する必要があります。それは設定される列値の 1 つである必要があります。

- additional - LDAP ディレクトリーのスキーマに定義された各エントリーには追加列が 1 つまたは複数必要である場合があります。これは LDAP サーバー実装によるため、LDAP のドキュメントを確認してください。

クエリーの例を以下に示します。

```
SELECT * FROM LdapModel.People
```

```
INSERT INTO LdapModel.People (dn, sn,
objectclass, Name) VALUES
('cn=JoeYoung,ou=people,dc=example,dc=org','Young','person',
'Joe Young')
```

```
UPDATE LdapModel.People SET
PhoneNumber='(314) 299-2999' WHERE
DN='cn=JoeYoung,ou=people,dc=example,dc=org'
```

```
DELETE FROM LdapModel.People WHERE
DN='cn=JoeYoung,ou=people,dc=example,dc=org'
```

## 9.15. XSD スキーマファイル

XML スキーマ (XSD) ファイルをインポートするには、以下の手順に従います。

1. In Model Explorer, click the **File > Import** action  in the toolbar or select a project, folder or model in the tree and choose Import...
2. インポートオプション **TTeiid Designer > XML Schemas** を選択し、**Next>** をクリックします。
3. **Import XSD Schemas from file system** または **Import XSD Schemas via URL** を選択し、**Next >** をクリックします。
4.
  - a. ファイルシステムからインポートする場合は、Import XSD Files ダイアログが表示されます。**Browse** ボタンをクリックし、インポートする XSD ファイルが含まれるディレクトリーを見つけます。
    - ※ ディレクトリーの XSD ファイルをすべて選択するには、左パネルのフォルダーの横にあるチェックボックスをクリックします。
    - ※ XSD ファイルを個別に選択するには、右パネルでファイルの横にあるチェックボックスをクリックします。





図9.62 ファイルシステムからの XSD の選択

- b. URL からインポートする場合は、**Import XML Schemas via URL** オプションを選択して **OK** をクリックし、最終の **Add XML Schema URLs** ウィザードページを表示します。

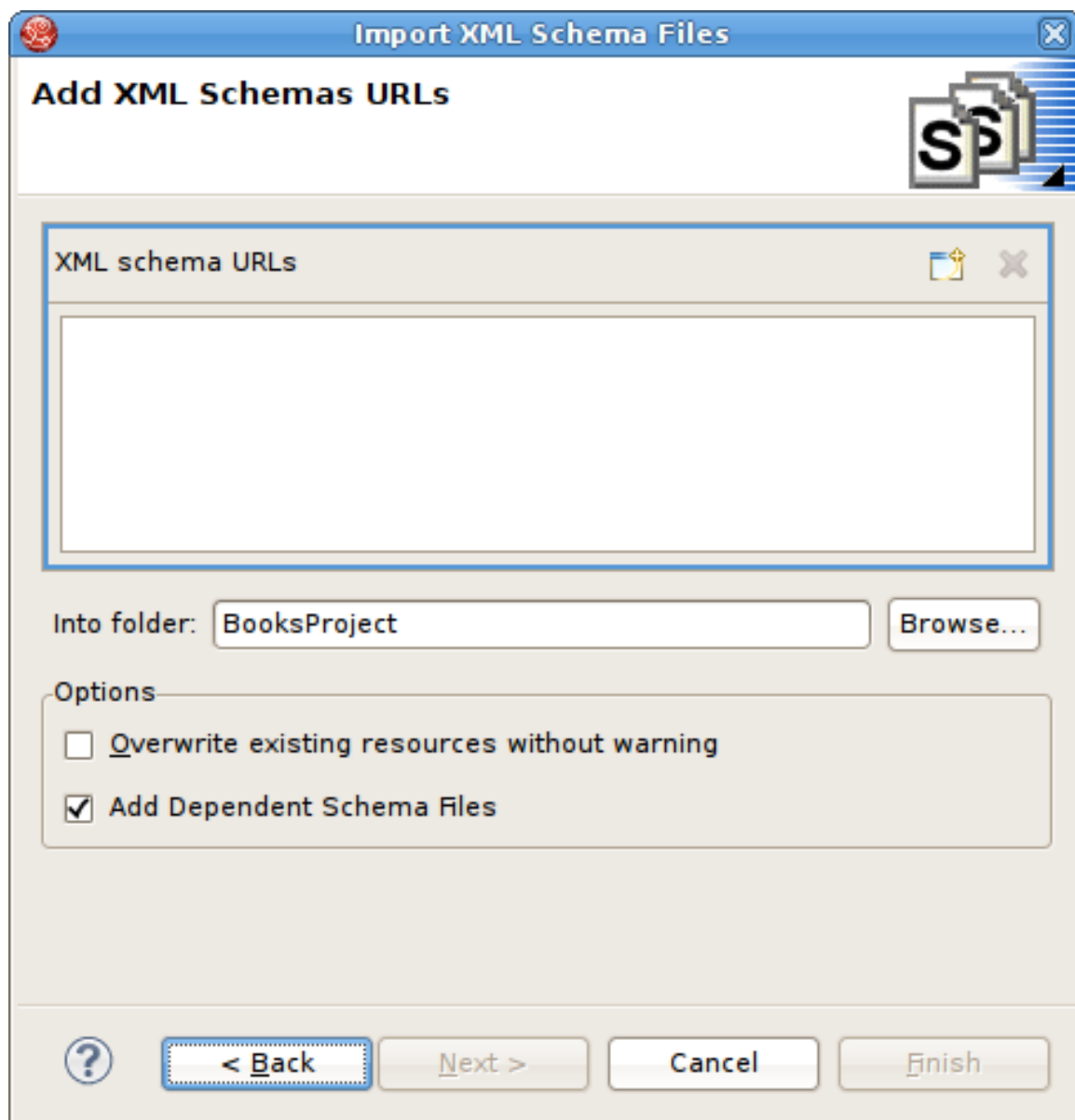


図9.63 Add XML Schema URLs ダイアログ

5. Click the **Add XML Schema URL** button . Enter a valid schema URL. Click **OK**. Schema will be validated and resulting entry added to the list of XML Schema URLs.

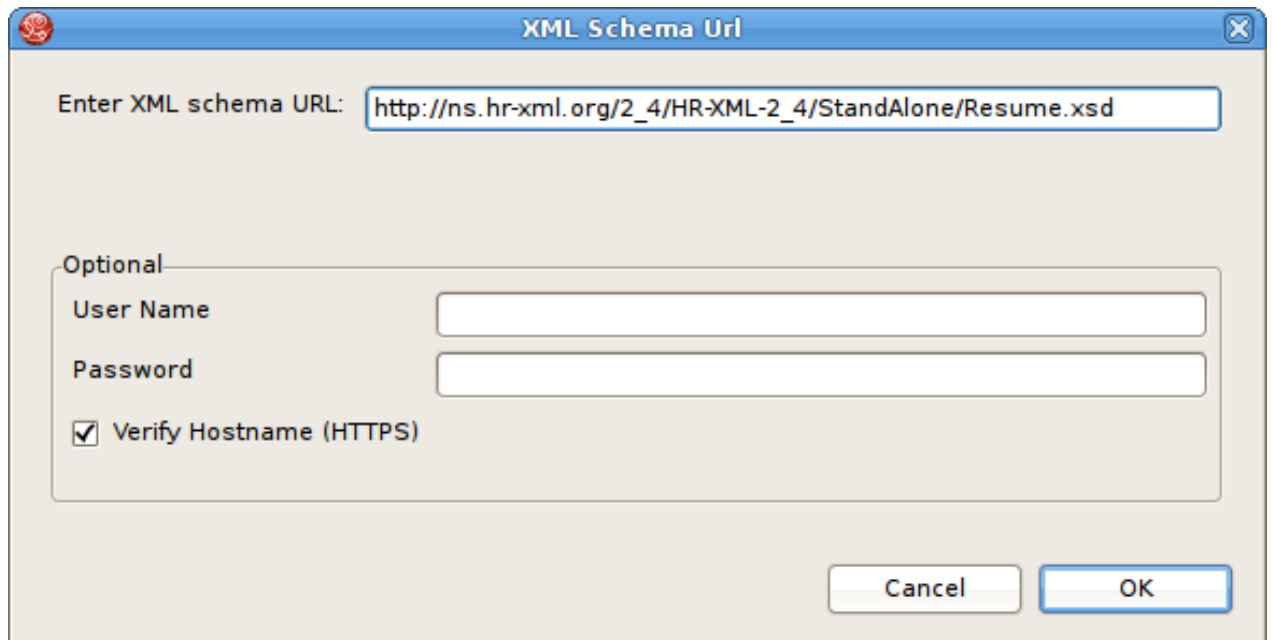
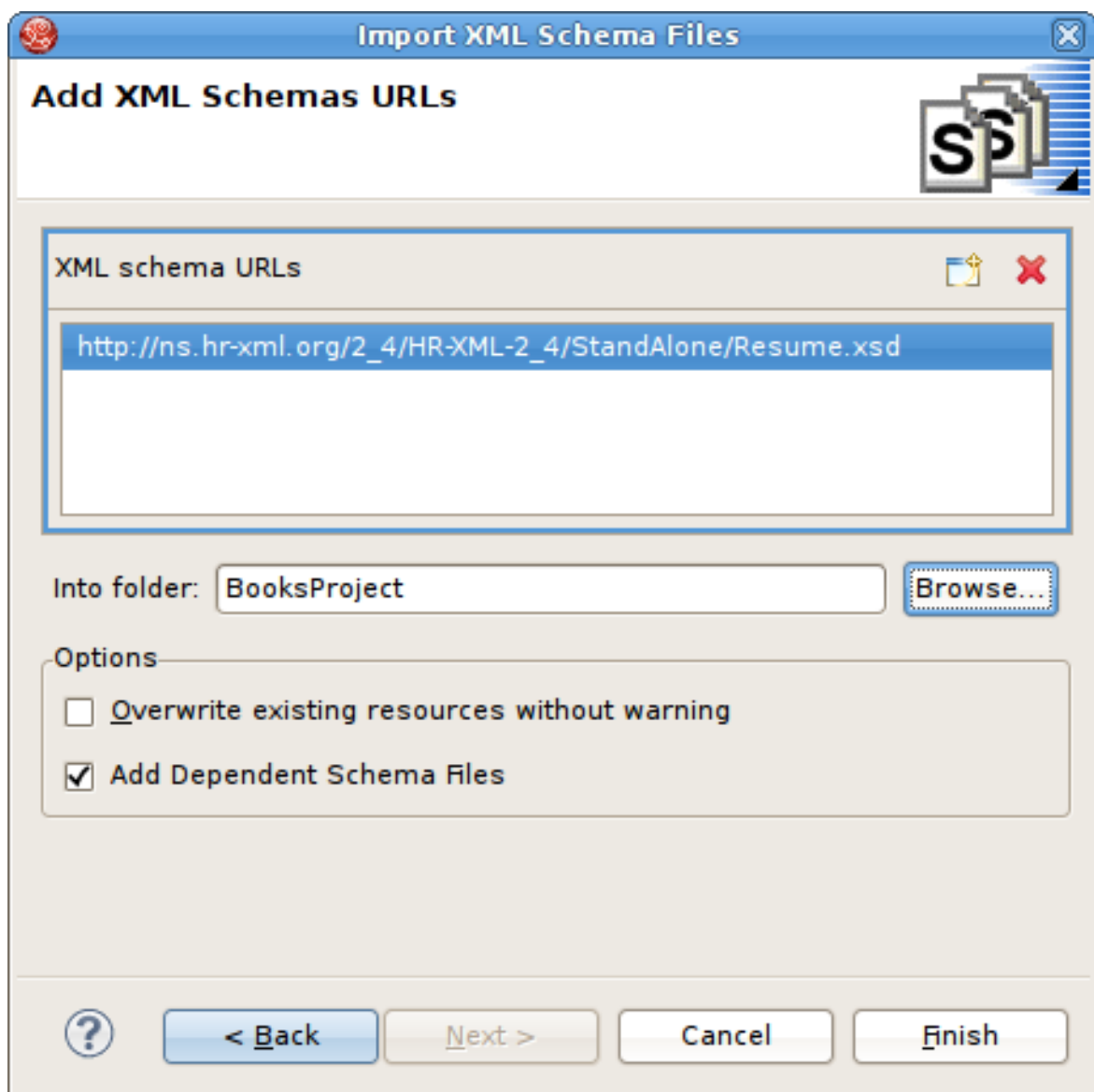


図9.64 XSD スキーマ URL の追加

スキーマ URL が XML スキーマ URL リストに表示されます。



## 図9.65 XSD スキーマ URL の追加

6. **Finish** をクリックします。



### 注記

XSD ファイルが依存するファイルが存在する場合があります。「Add Dependent Schema Files」を選択すると、このインポーターは依存関係を判断し、依存関係もインポートします。

## 9.16. Red Hat JBoss Data Grid と複雑なデータタイプ



### 注記

Teiid Designer を使用して pojo のビューのリバースエンジニアリングを行うと、BigDecimal データタイプがビューで定義されます。シリアル化に使用される Google Protobuf の場合、残念ながら複雑なデータタイプは C または C++ に変換することができません。このため、プリミティブなデータタイプのみを使用することが推奨されます。(複雑なデータタイプを含むビューを実体化する場合、もしくは複雑なデータタイプを含む POJO を格納している既存の JDG キャッシュがある場合、この状況が発生します。)

protobuffer は BigDecimal を直接サポートしないので、以下の 3 つのオプションがあります。

1. すべてのプリミティブデータタイプを使用する。
2. 変換を処理するマーシャラーを実装する。その場合、.proto ファイルも作成する必要があります。(ファイル作成については、Red Hat JBoss Data Grid を参照してください。)
3. BigDecimal を文字列に変換するビューを作成し、そのビューを実体化する。

## 第10章 モデルオブジェクトの作成および編集

### 10.1. 新しいモデルオブジェクトの作成

#### 10.1.1. 新しいモデルオブジェクトの作成

本項では、モデルに含まれた既存のモデルオブジェクトを作成および編集する **Teiid Designer** の機能について説明します。**Teiid Designer** は、さまざまなタイプのメタデータをモデル化するフレームワークを提供します。各メタモデルタイプは、どこに何を作成できるかを制約する親子の関係のセットを持ちます。たとえば、保存されたプロシージャに列属性を作成することはできず、Web サービス演算の出力メッセージでマッピングクラスを作成することもできません。

**Teiid Designer** は、これらのモデルの新しい子と孫を作成する共通のアクションのセットを提供します。

以下のアクションを使用すると、モデルエクスプローラービュー、ダイアグラムエディター、またはテーブルエディターで新しいモデルオブジェクトを直接作成できます。

- New Child アクション
- New Sibling アクション
- New Association Action (新規の関連アクション)

#### 10.1.2. New Child アクション

モデルエクスプローラービューでモデルオブジェクトの子を新たに作成するには、以下を行います。

1. 子を追加する親オブジェクトを選択します。たとえば、パッケージをパッケージに追加したり、属性をクラスに追加したりすることができます。
2. コンテナオブジェクトを右クリックします。ポップアップメニューの **New Child** をクリックします。その後、追加する子オブジェクトを選択できます。

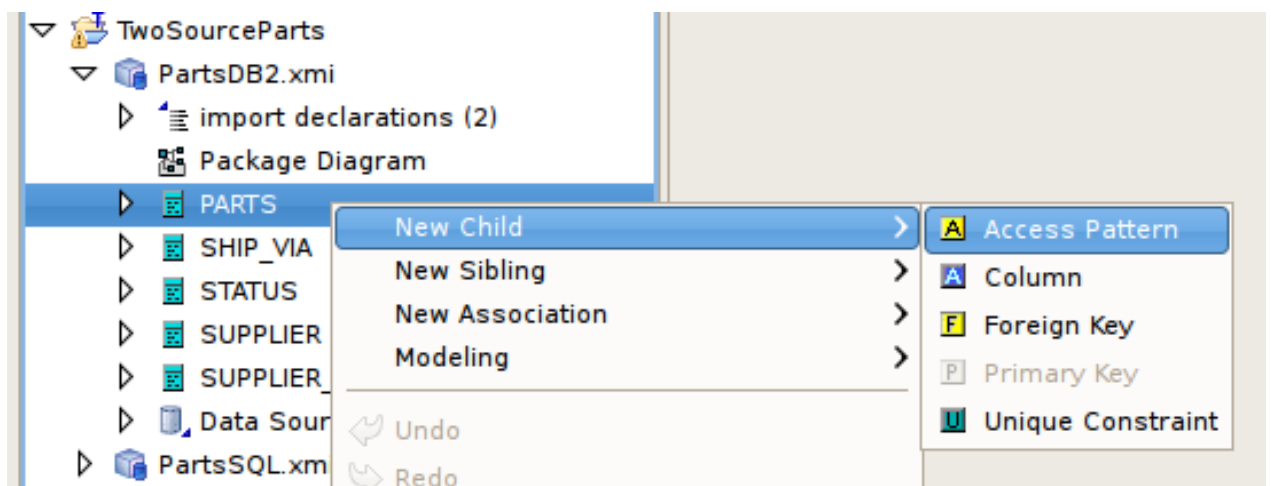


図10.1 モデルエクスプローラーの New Child アクション

3. 新しいモデルオブジェクトがモデルエクスプローラービューに表示され、名前を変更するためにハイライトされます。

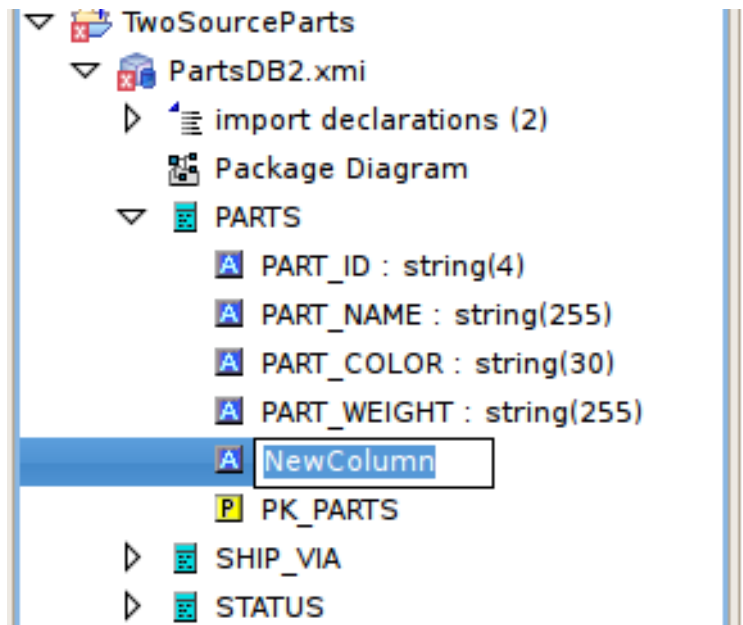


図10.2 エクスプローラーの新しいモデルオブジェクト

ダイアグラムエディターでモデルオブジェクトの子を新規作成するには、以下を行います。

1. 子を追加する親オブジェクトを選択します。たとえば、パッケージをパッケージに追加したり、属性をクラスに追加したりすることができます。
2. コンテナオブジェクトを右クリックします。ポップアップメニューの **New Child** をクリックします。その後、追加する子オブジェクトを選択できます。

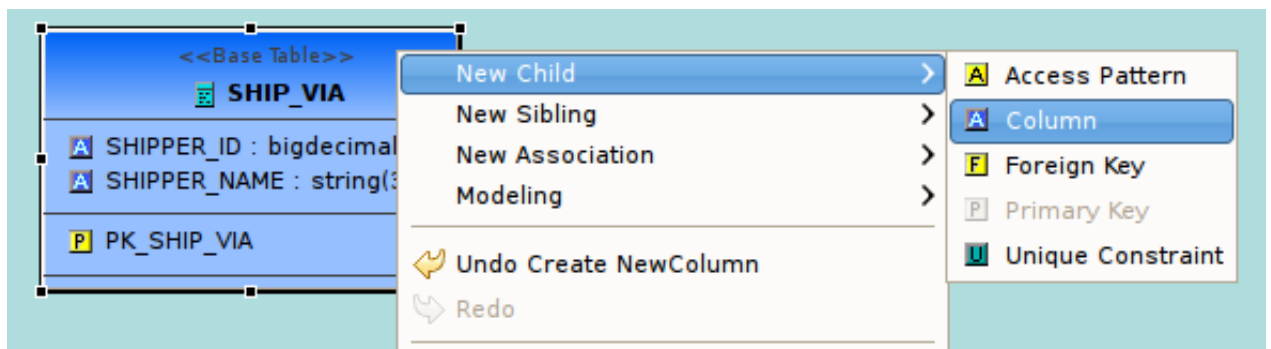


図10.3 ダイアグラムでの New Child アクション

3. 新しいモデルオブジェクトがダイアグラムに表示され、名前を変更するためにハイライトされます。

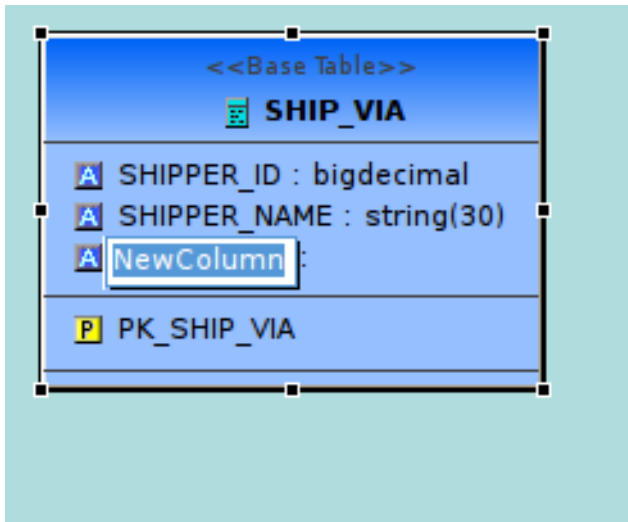


図10.4 ダイアグラムの新しいモデルオブジェクト

テーブルエディターでモデルオブジェクトの子を新規作成するには、以下を行います。

1. 子を追加する親オブジェクトの行を選択します。たとえば、列を追加する場合は **Base Table** タブをクリックし、ベーステーブル行を選択します。
2. テーブル行を右クリックします。ポップアップメニューの **New Child** をクリックします。その後、追加する子オブジェクトを選択できます。

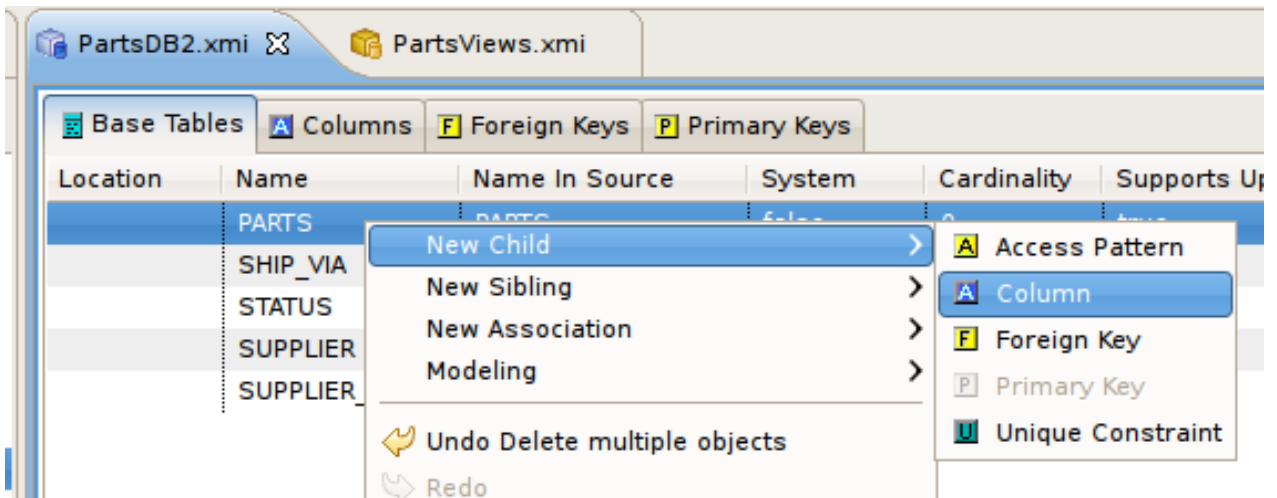


図10.5 テーブルエディターの New Child アクション

3. テーブルエディターで選択されたタブが子オブジェクトタイプのタブに変更されます。新しいモデルオブジェクト行が表示され、名前を変更するために行の名前テーブルセルがハイライトされます。

### 10.1.3. New Sibling アクション

モデルエクスプローラービューでモデルオブジェクトの兄弟を新たに作成するには、以下を行います。

1. 兄弟を追加するオブジェクトを選択します。たとえば、列に列の兄弟を追加できます。
2. そのオブジェクトを右クリックします。ポップアップメニューの **New Sibling** をクリックします。その後、追加する兄弟オブジェクトを選択できます。



図10.6 モデルエクスプローラーの New Sibling アクション

3. 新しいモデルオブジェクトがモデルエクスプローラービューに表示され、名前を変更するためにハイライトされます。

ダイアグラムエディターでモデルオブジェクトの兄弟を新規作成するには、以下を行います。

1. 兄弟を追加するオブジェクトを選択します。たとえば、列に列の兄弟を追加できます。
2. そのオブジェクトを右クリックします。ポップアップメニューの **New Sibling** をクリックします。その後、追加する兄弟オブジェクトを選択できます。

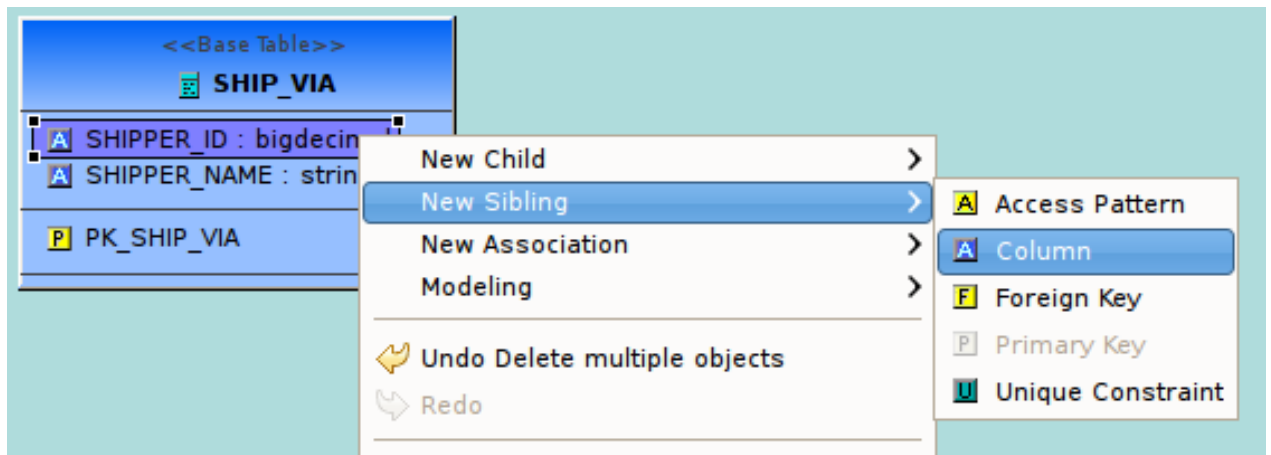


図10.7 ダイアグラムでの New Sibling アクション

3. 新しいモデルオブジェクトがダイアグラムに表示され、名前を変更するためにハイライトされます。

テーブルエディターセクションでモデルオブジェクトの兄弟を新規作成するには、以下を行います。

1. 兄弟を追加するオブジェクトの行を選択します。たとえば、列に列の兄弟を追加できます。
2. 行を右クリックします。ポップアップメニューの **New Sibling** をクリックします。その後、追加する兄弟オブジェクトを選択できます。



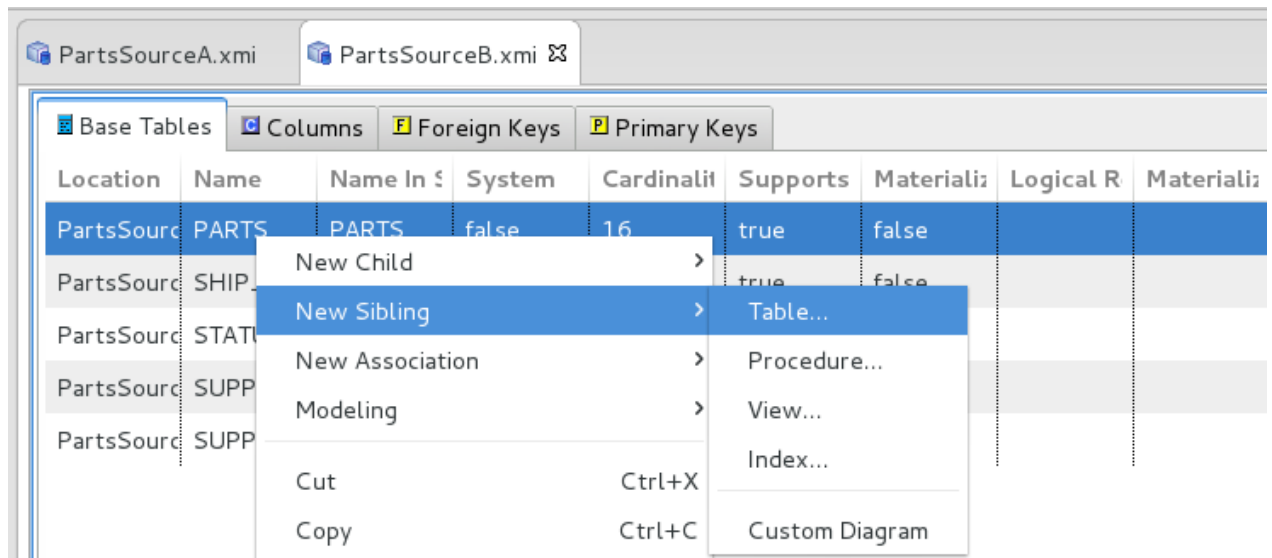


図10.8 テーブルエディターの New Sibling アクション

3. テーブルエディターで選択されたタブが子オブジェクトタイプのタブに変更されます。新しいモデルオブジェクト行が表示され、名前を変更するために行の名前テーブルセルがハイライトされます。

#### 10.1.4. New Association Action (新規の関連アクション)

モデルエクスプローラービューでモデルオブジェクト間の関係を新規作成するには、以下を行います。

1. 関連付ける 2 つのオブジェクトを選択します。たとえば、異なるベーステーブルの列を選択します。
2. 右クリックします。ポップアップメニューで **New Association > Foreign Key Relationship** とクリックします。

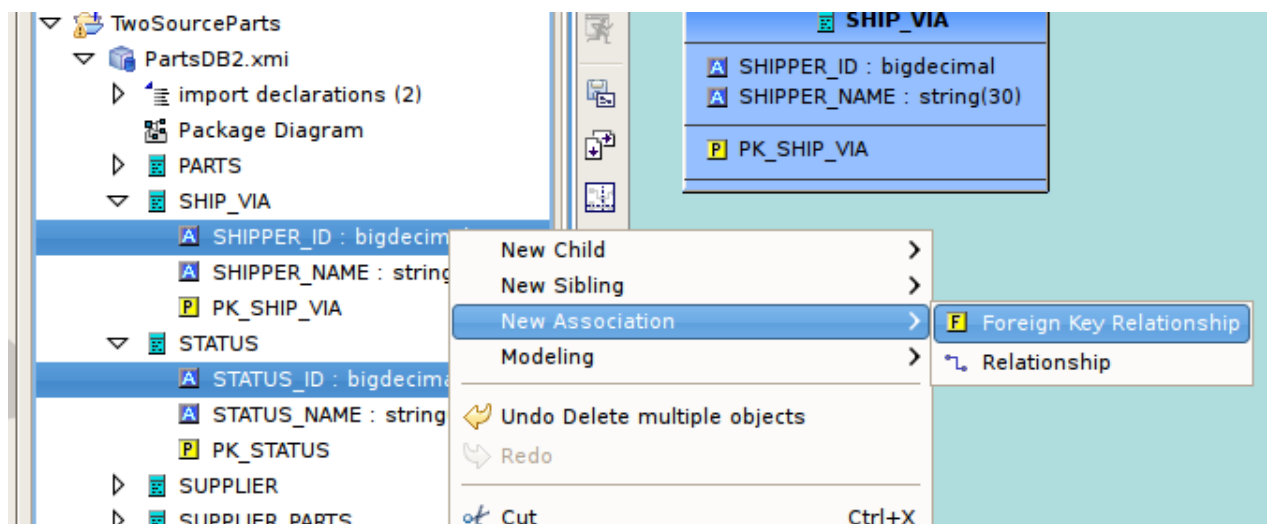


図10.9 モデルエクスプローラーの New Association アクション

3. ダイアグラムに新しい関係のリンクが表示されます。

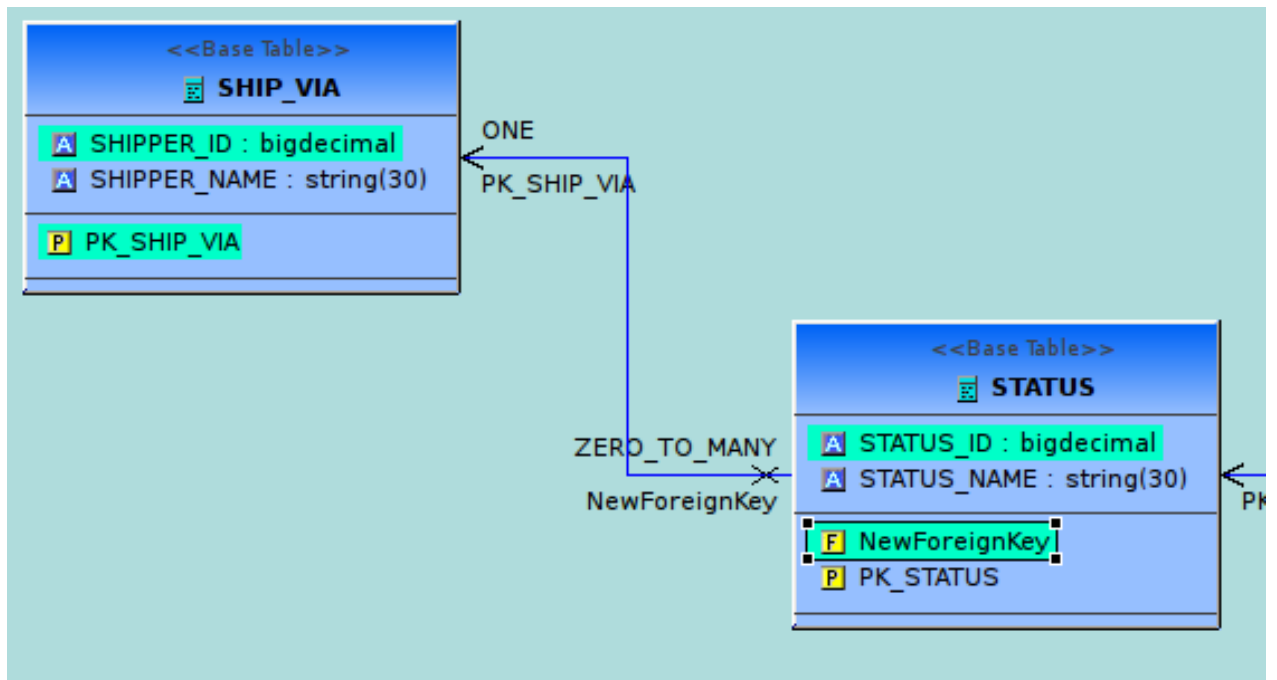


図10.10 ダイアグラムの新しい関係

ダイアグラムエディターでモデルオブジェクト間の関係を新規作成するには、以下を行います。

- ※ A.
  - a. 関連付ける 2 つのオブジェクトを選択します。たとえば、異なるベーステーブルの列を選択します。
  - b. 右クリックします。ポップアップメニューで **New Association > Foreign Key Relationship** とクリックします。
  - c. ダイアグラムに新しい関係のリンクが表示されます。Column、Foreign Key、および Primary Key 参照プロパティーが選択された列、新しいプライマリーキー、および新しい外部キーに適切に設定されます。
- B.
  - a. テーブルで列を選択します。
  - b. 列を別のテーブルにドラッグします。列をこの列にドラッグおよびドロップします。ドロップ先の列は黄色でハイライトされます。
  - c. ダイアグラムに新しい関係のリンクが表示されます。Column、Foreign Key、および Primary Key 参照プロパティーが選択された列、新しいプライマリーキー、および新しい外部キーに適切に設定されます。

テーブルエディターでモデルオブジェクト間の関係を新規作成するには、以下を行います。

1. 関連付ける 2 つのオブジェクトを選択します。たとえば、異なるベーステーブルの列を選択します。
2. 右クリックします。ポップアップメニューで **New Association > Foreign Key Relationship** と選択します。
3. 新しい Foreign Key および Primary Key オブジェクトは、テーブルエディターのそれぞれに対応する内容に追加されます。Column、Foreign Key、および Primary Key 参照プロパティーが選択された列、新しいプライマリーキー、および新しい外部キーに適切に設定されます。

### 10.1.5. 新規モデルオブジェクトのウィザード

Teiid Designer は、ビューおよびソースリレーショナルモデルの子 (テーブル、ビュー、プロシージャ、およびインデックス) の **New Child** および **New Sibling** ウィザードを提供します。

- ※ **Create Relational Table Wizard** では、列、一意キー、外部キーの定義およびインデックスを含む完全なリレーショナルテーブルを作成できます。



### 注記

リレーショナルビューウィザードはリレーショナルテーブルウィザードと全く同じです。

- ※ **Create Relational Procedure Wizard** では、列、一意キー、外部キーの定義およびインデックスを含む完全なリレーショナルプロシージャを作成できます。リレーショナルプロシージャオブジェクトは、標準的なプロシージャやソース関数 (プッシュダウン関数) を含む異なるタイプのプロシージャを表すことができます。 **New Child - Procedure...** アクションが開始されると、最初のダイアログでプロシージャタイプをオプションで選択できます。



### 注記

ユーザー定義の関数をビューモデルでモデル化できます。

ウィザードの 2 つ目のダイアログでは、選択したタイプに基づいて **Create Relation Procedure** ダイアログをカスタマイズします。

ソース関数は、データベースにすでにデプロイされ、アクセス可能になっているプロシージャです。リレーショナルモデルのソース関数を定義すると、変換 SQL 内からこれらの関数を呼び出すことができ、関数は実行のためにデータベースにプッシュされます。

- ※ **Create Relational Index Wizard**: インデックスはリレーショナルテーブルオブジェクトと同時に作成できます。また **New Child - Index...** アクションおよびダイアログでも作成できます。
- ※ **Create View Model Objects Wizards**: ビューモデルでは、テーブル、プロシージャ (標準のプロシージャおよびユーザー定義の関数) およびインデックスのみを作成できます。ビューテーブルおよびプロシージャでは、ウィザードの主な違いは SQL 変換タブが含まれていることです。
- ※ **Create User Defined Functions Wizard**: カスタム関数をタイプ「function」のビューモデルプロシージャとして定義できます。ビューモデルに対して **New Child - Procedure...** と選択すると、最初のダイアログで標準のビュープロシージャまたはユーザー定義の関数のいずれかが含まれるプロシージャタイプをオプションで選択できます。

ユーザー定義の関数には、Java クラス、メソッド、コードが含まれる jar ファイルへのパスなどの追加のプロパティが必要になります。

## 10.2. Model Object Editors

モデルオブジェクト編集の主なアクションは次のとおりです。

- ※ **Cut** - 選択したオブジェクトを削除し、クリップボードへコピーします。
- ※ **Copy** - 選択したオブジェクトをクリップボードへコピーします。
- ※ **Paste** - クリップボードの内容を選択したコンテキストに貼り付けます。

- ※ Clone - 選択したオブジェクトの複製を同じ名前で同じ場所に作成します。ユーザーはツリーで新しいオブジェクトの名前を変更できます。
- ※ Delete - 選択したオブジェクトを削除します。
- ※ Rename - オブジェクトの名前を変更できます。

これらのアクションは Teiid Designer の **Edit menu** にあり、モデルエクスプローラー、ダイアログエディター、およびテーブルエディターで選択したモデルオブジェクトを右クリックすると表示されます。

### サブメニューのモデル化

オブジェクト作成の New Child、New Sibling、および New Association メニューの他に、Teiid Designer は実行できるオブジェクト固有のアクションを表す Modeling サブメニューを提供します。

たとえば、ソーステーブルを選択すると、以下の Modeling メニューが表示されます。



図10.11 ソーステーブルの Modeling サブメニュー

ビューテーブルを選択すると、メニューは仮想操作に関連するアクションを反映します。

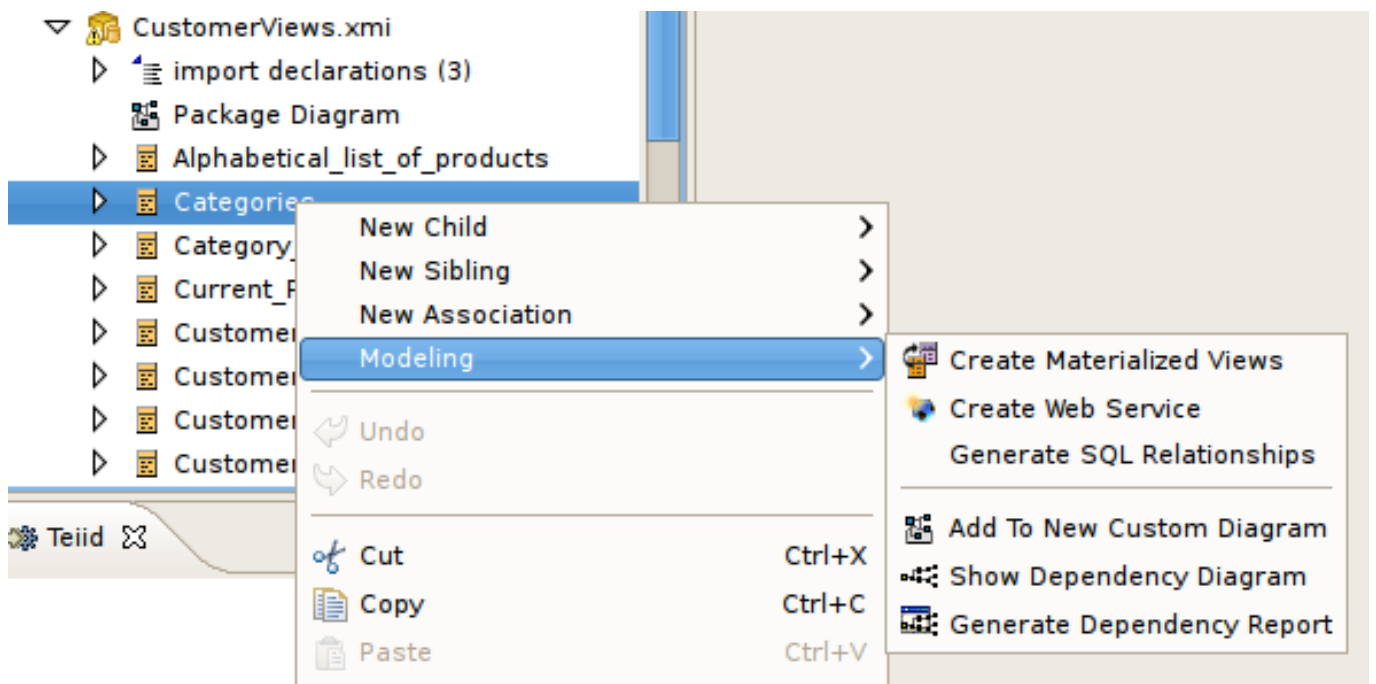


図10.12 ソーステーブルの Modeling サブメニュー

Teiid Designer は、複雑なモデルオブジェクトやそれらの一意なプロパティを処理する特別なオブジェクトエディターも提供します。

## 10.3. Transformation Editor

### 10.3.1. Transformation Editor

Teiid Designer の **Transformation Editor** では、物理メタデータソースまたはその他の仮想メタデータから仮想メタデータ情報を派生させる方法と、ソースの更新方法を記述するクエリトランスフォーメーションを作成できます。

**Transformation Editor** は、SQL クエリーの作成に使用できる複数のツールを提供します。ツールを使用したり、SQL クエリーを **Transformation Editor** に入力したりすることができます。

トランスフォーメーションを編集するには、以下を実行します。

- ▶ **Edit** をダブルクリックします。
  - **Model Explorer** または **Diagram Editor** のリレーショナルビューテーブルまたはプロシージャ
  - トランスフォーメーションダイアグラムまたはマッピングトランスフォーメーションダイアグラムのトランスフォーメーションノード
- ▶ **Model Explorer**、**Diagram Editor**、または **Table Editor** で選択したオブジェクトを右クリックし、**Edit** を選択します。
  - リレーショナルビューテーブルまたはプロシージャ
  - トランスフォーメーションダイアグラムまたはマッピングトランスフォーメーションダイアグラムのトランスフォーメーションノード
  - マッピングダイアグラムまたはマッピングトランスフォーメーションダイアグラムのマッピングクラス

選択したオブジェクトのモデルに対して **Model Editor** が開かれていない場合は **Model Editor** が開かれます。

**Diagram Editor** で対応するトランスフォーメーションダイアグラムが開かれた後、**Diagram Editor** の下部に **Transformation Editor** が表示されます。

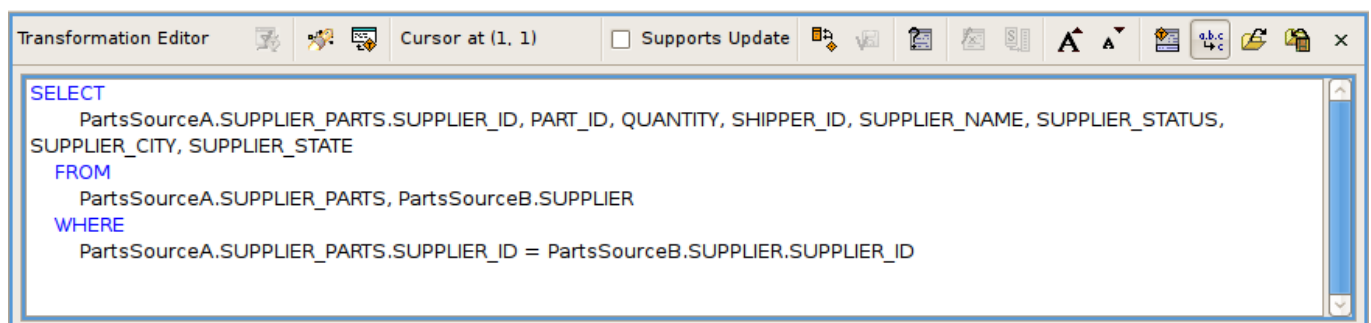


図10.13 String プロパティの編集




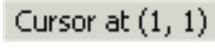




この仮想クラスが更新をサポートする場合、**Transformation Editor** の下部にあるタブを使用すると仮想クラスがサポートするクエリーの各タイプに SQL を入力できます。この仮想クラスが更新をサポートしない場合は、SELECT タブのみを使用できます。










クエリーのタイプに対応するため、使用できる各タブで個別の SQL クエリーを入力することができます。

Transformation Editor 内では以下を実行できます。

- ✦ この仮想クラスで特定の更新トランスフォーメーションタイプを無効にします。
- ✦ 提供された SQL テンプレートでトランスフォーメーションを開始します。
- ✦ トランスフォーメーションで使用する criteria 句をビルドまたは編集します。
- ✦ トランスフォーメーションで使用する式をビルドまたは編集します。
- ✦ トランスフォーメーション内で文字列を見つけ、置き換えます。
- ✦ 内容にエラーがないよう、トランスフォーメーションを検証します。
- ✦ トランスフォーメーションにあるシンボルが仮想メタデータクラスの属性と一致するよう、属性を照合します。

Transformation Editor の表示の調整を設定することもできます。

- ✦ **Transformation Editor** ツールバーアクションの概要を以下に示します。
  -  Preview Virtual Data - executes a simple preview query for the target table or procedure of the transformation being edited.
  -  Search Transformations - provides a simple way select and edit another transformation based SQL text search criteria.
  -  Edit Transformation - provides a simple way to change which transformation to edit without searching in a diagram or the Model Explorer. Click the action and select from a list of views, tables, procedures or operations from the currently edited model.
  -  **Cursor at (1, 1)** Cursor Position (line, column) - shows the current line and column position of the insertion cursor. For example, Cursor Position(1,4) indicates that the cursor is presently located at column 4 of line 1.
  -  **Supports Update** Supports Update - checkbox allows you to enable or disable updates for the current transformation target. If Supports Update is selected, the editor shows four tabs at the bottom for the Select, Update, Insert and Delete transformations. If Supports Update is cleared, all updates are disabled and only the Select transformation is displayed.
  -  Reconcile - allows you to resolve any discrepancies between the transformation symbols and the target attributes. Clicking this button will display the Reconcile Virtual Target Attributes dialog box in which you can resolve discrepancies.
  -  Save/Validate - saves edits to the current transformation and validates the transformation SQL. Any Warning or Error messages will be displayed at the bottom of the editor in the messages area. If the SQL validates without error, the message area is not displayed.
  -  Criteria Builder - allows you to build a criteria clause in your transformation. The button will enable if the cursor position is within a query that allows a criteria. Pressing the button will launch the Criteria Builder dialog. If the Criteria Builder is launched inside an existing criteria, that criteria will be displayed for edit, otherwise the Criteria Builder will be initially empty.

-  Expression Builder - allows you to build an expression within your transformation. The button will enable if the cursor position is at a location that allows an expression. Pressing the button will launch the Expression Builder dialog. If the Expression Builder is launched inside an existing expression, that expression will be displayed for edit, otherwise the Expression Builder will be initially empty.
  -  Expand Select \* - allows you to expand a SELECT \* clause into a SELECT clause which contains all of the SELECT symbols. The button will enable only if the cursor is within a query that contains a SELECT \* clause that can be expanded.
  -  Increase Font Size - increases the font size of all editor text by 1.
  -  Decrease Font Size - decreases the font size of all editor text by 1.
  -  Show/Hide Messages - toggles the display of the message area at the bottom of the transformation editor.
  -  Optimize SQL - when toggled ON, will use the short names of all SQL symbols that can be optimized. Some symbol names may remain fully qualified in the event of a duplicate name or if the optimizer is unable to optimize it. When the action is toggled OFF, all symbol names will be fully qualified.
  -  Import SQL Text - allows you to import an SQL statement from a text file on your file system. Pressing this button will display an import dialog in which you can navigate to the file.
  -  Export SQL Text - allows you to export the currently displayed SQL statement into a text file on your file system. Pressing this button will display an export dialog in which you can choose the location for export.
  -  Close X - Transformation Editor を閉じます。
- \* **Transformation Editor** コンテキストメニューを表示するには、エディターのテキストエリア内でマウスの右ボタンをクリックします。コンテキストメニューを以下に示します。

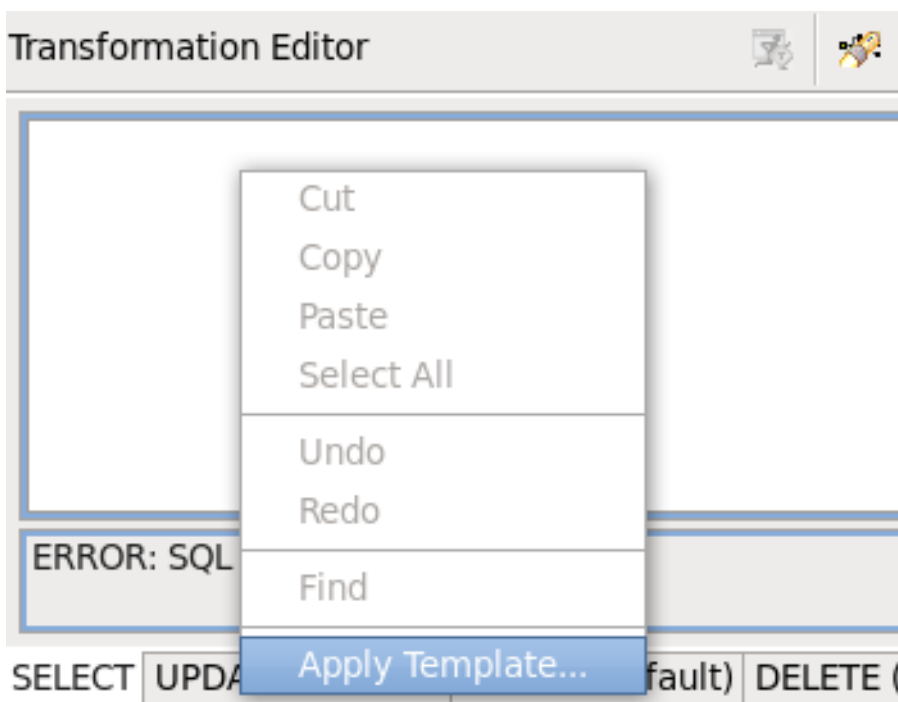


図10.14 Transformation Editor コンテキストメニュー

コンテキストメニューアクションの概要は次のとおりです。

- Cut - Copy - Paste - エディター内でテキストを切り取り、コピー、または貼り付けする一般的なテキストエディターアクション。
- Undo - Redo - 前回のアクションを取り消しまたは再度実行します。
- Find - トランスフォーメーション内でテキストを検索および置き換えできる Find and Replace ダイアログを表示します。
- Apply Template... - 一般的な SQL パターンのリストから 開始 SQL テンプレートを選択できる Choose an SQL Template ダイアログを表示します。このダイアログの説明は View Table ウィザードの項を参照してください。

### 10.3.2. Reconciler の使用

Transformation Editor の Reconciler は、Target View 属性とトランスフォーメーション SQL を一致させる迅速およびグラフィカルな方法を提供します。変更を加えると、ダイアログの上部に全体の状況が表示され、正常に編集を行えるようにします。

To launch the Reconciler, click the **Reconcile Transformation** button  in the Transformation Editor.

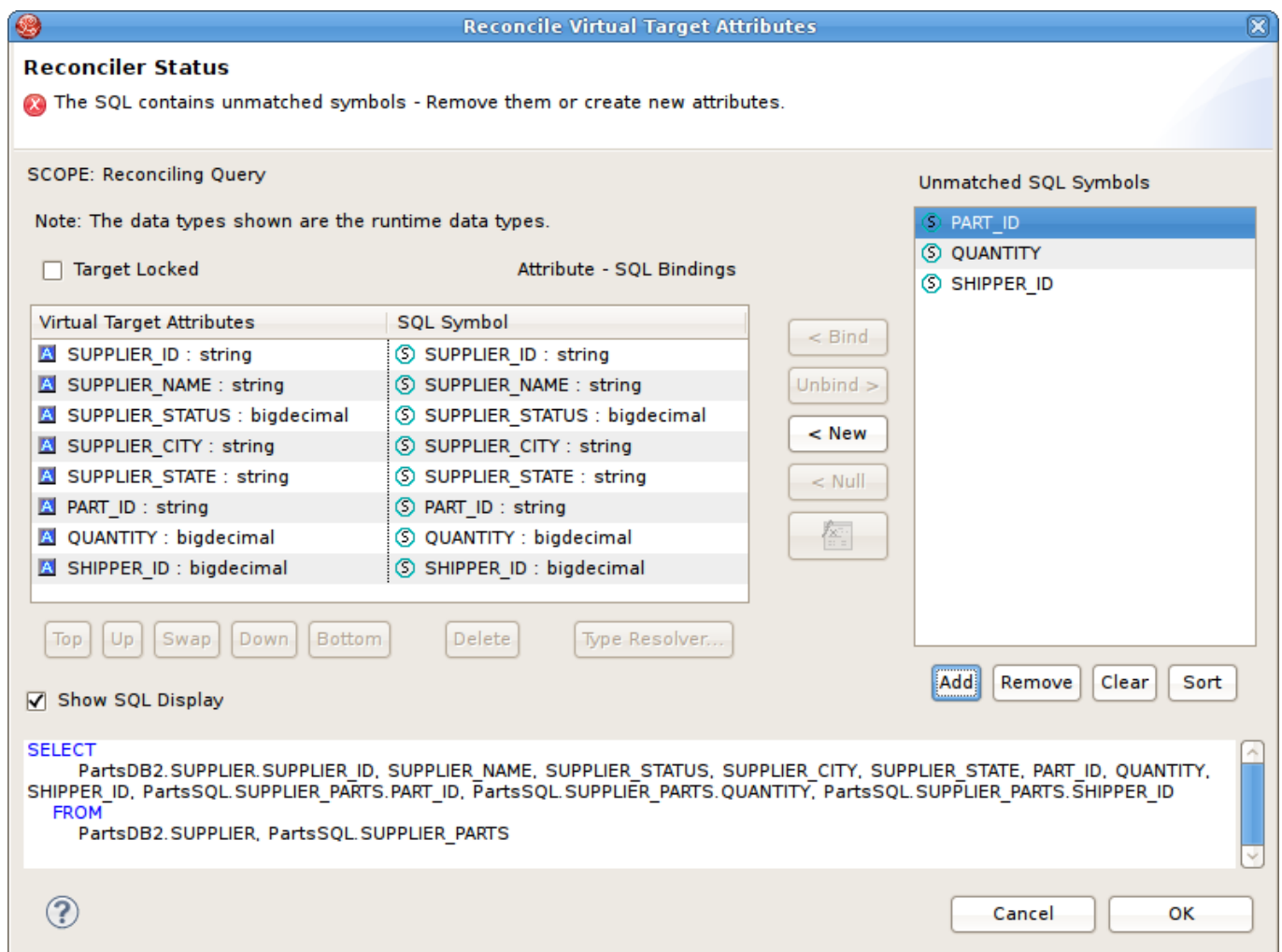


図10.15 Reconciler ダイアログ

ダイアログの異なるセクションの概要は次のとおりです。



- ※ Target Attributes - SQL Symbol Table: このテーブルは、左列のターゲット属性と右列の SQL シンボルを表示します。SQL シンボルは SQL トランスフォーメーションから射影されたシンボルです。属性の横に表示されるとシンボルはターゲット属性にバインドされたものとみなされます。

ターゲット属性がバインドされていないと、行が赤でハイライトされます。すべての属性が対応する SQL シンボルのバインディングを持つまでトランスフォーメーションは無効です。

以下をテーブルセクションで実行できます。

- ターゲット属性のロック: ターゲット属性の順序付けをロックするには、Lock Target Attributes チェックボックスを選択します。これにより、属性がロックされます。
- 属性の再順序: テーブル直下の Top、Up、Swap、Down、および Bottom コントロールを使用します。単一または複数のテーブル行を選択し、希望のアクションボタンをクリックします。
- 属性の削除: ターゲット属性を 1 つまたは複数削除するには、削除するテーブル行を選択し、Delete ボタンをクリックします。
- タイプの解決: 属性と SQL シンボルのバインディングにデータタイプの競合がある場合、メッセージが表示されます。データタイプの競合を解決するために、Datatype Resolver Dialog が提供されます。テーブル行をクリックし、Type Resolver... ボタンをクリックしてダイアログを表示します。詳細は、「データタイプリゾルバーの使用」を参照してください。
- ※ 一致しない SQL シンボルのリスト: このリストは属性とシンボルのバインディングテーブルの右にあり、ターゲットテーブル属性にバインドされないトランスフォーメーション SQL から SQL シンボルを表示します。

以下をリストセクションで実行できます。

- SQL シンボルの追加: SQL シンボルをリストに追加するには、Add ボタンをクリックします。トランスフォーメーションソーステーブルで使用できるシンボルがすべて表示されるダイアログが表示されます。追加したいシンボルをクリックし、OK をクリックします。
- シンボルの削除または消去: SQL シンボルを 1 つまたは複数削除するには、リスト項目を選択してから Remove ボタンをクリックします。SQL シンボルリスト全体を消去するには、Clear ボタンをクリックします。
- シンボルのソート: デフォルトでは、SQL クエリーが表示される順番にシンボルが表示されます。リストでアルファベット順に表示するには、Sort ボタンをクリックします。
- ※ バインディングの制御: Binding Controls は Attribute-Symbol テーブルと Unmatched SQL Symbols リストの間にあります。これらのボタンを使用して、属性とシンボルのバインディングを定義します。

以下をバインディングの制御で実行できます。

- Bind: このボタンは SQL シンボルをターゲット属性にバインドします。一致しない SQL シンボルとターゲット属性を選択し、Bind をクリックしてバインディングを確立します。
- Unbind: このボタンは Attribute-Symbol バインディングを解除します。すでにバインドされている属性をテーブルで選択し、Unbind をクリックします。SQL シンボルが Unmatched Symbols リストに出力されます。
- New: このボタンは、一致しない SQL シンボルを使用して新しいターゲット属性を作成します。リストから一致しない SQL シンボルを選択し、New をクリックします。新しいターゲット属性が Attribute-Symbol テーブルの下部に追加され、選択された SQL シンボルにバインドされます。
- Null: このボタンは、SQL シンボルをバインドする代わりに null をターゲット属性にバインドします。Attribute-Symbol テーブルの列を選択し、Null をクリックします。ターゲット属性は null にバインドされます。最初に SQL シンボルにバインドされていた場合は、シンボルが Unmatched Symbol リストに出力されます。

- **Function:** このボタンを使用すると、バインディングで SQL シンボルの代わりに式を定義できます。式を定義するには、Attribute-Symbol テーブルの行を選択し、Function ボタンをクリックします。Expression Builder ダイアログが表示され、すべてのタイプの式を定義できます。Expression Builder の詳細は「Expression Builder の使用」を参照してください。
- ※ **SQL Display:** 現在のトランスフォーメーション SQL は Reconsiler ダイアログの下部に表示されます。SQL シンボルの追加または削除、およびその他の変更を行うと、その変更を反映して SQL Display が変更されます。OK をクリックすると、その SQL が新しいトランスフォーメーション SQL になります。必要な場合は、Show SQL Display チェックボックスの選択を解除し、SQL Display を非表示にすることができます。

バインディングの定義とデータタイプの解決を終了したら、OK をクリックして変更を許可します。編集を反映してトランスフォーメーション SQL が変更されます。

### 10.3.3. Datatype Resolver の使用

このダイアログへは **Reconciler** ダイアログからアクセスでき (「Reconciler の使用」を参照)、ターゲット属性とその SQL シンボルのデータタイプの競合を簡単に解決できます。データタイプバインディングの競合を解決するには、ソース SQL シンボルのデータタイプをすべて変換するか、ターゲット列のデータタイプをすべて変更します。

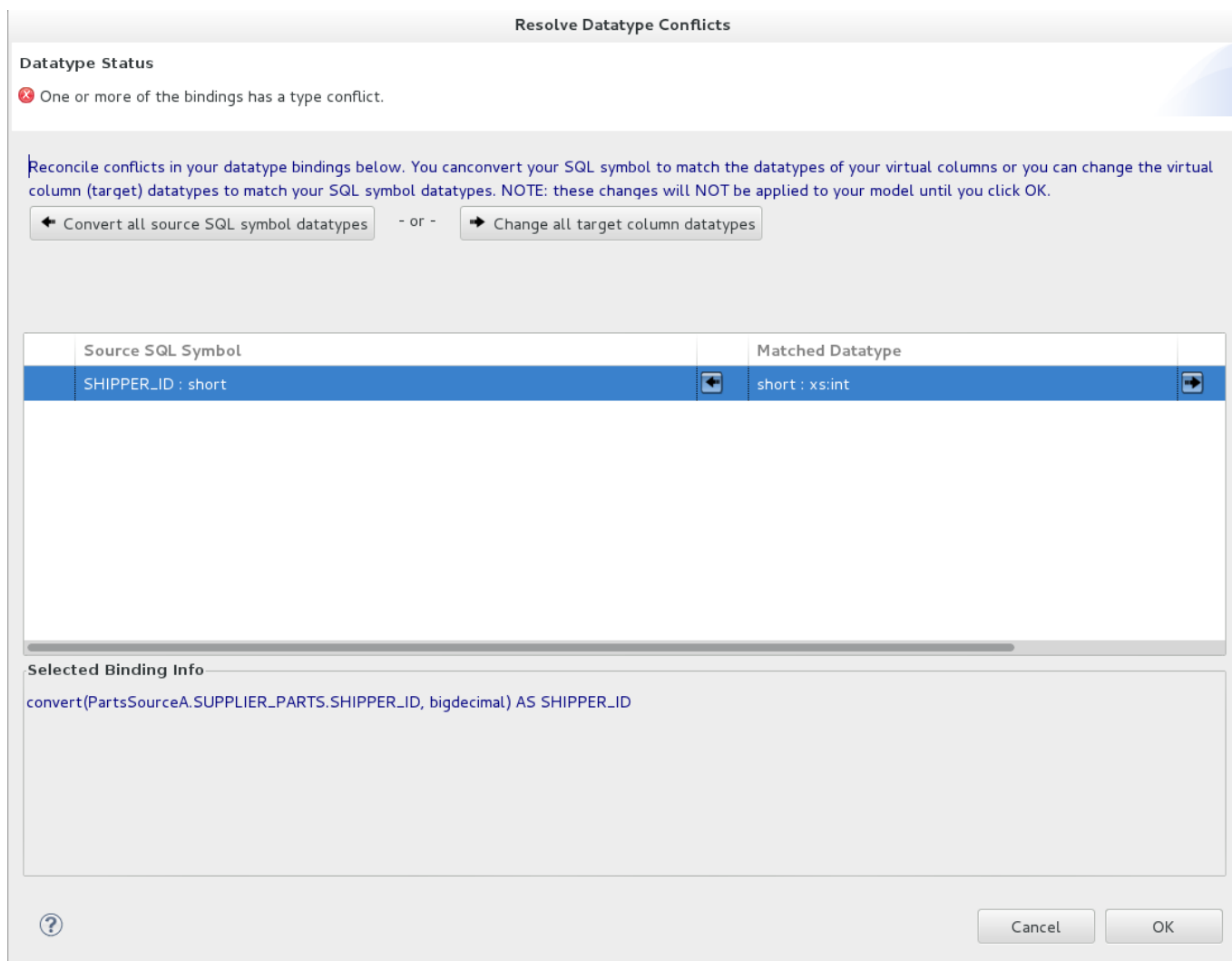


図10.16 Datatype Resolver ダイアログ

ダイアログの異なるセクションの概要は次のとおりです。

- ※ **Convert all source SQL symbol datatypes:** SQL シンボルのデータタイプが対象の属性のデータタイプに

対応するよう、このボタンをクリックして CONVERT 関数をテーブルのすべての SQL に適用します。

- ※ Change all target column datatypes: 提案されたデータタイプが適切でない場合は、このボタンをクリックし、Datatype ダイアログから他のデータタイプを選択します。
- ※ Source SQL Symbol - Matched Datatype Table: このテーブルには、選択したバインディングの SQL シンボルデータタイプの情報がすべて表示されます。テーブル行で選択し、下部パネルにデータを読み込みます。
- ※ Selected Binding Info: 下部パネルには選択した SQL シンボルのバインディング情報が表示されます。選択したバインディングの SQL シンボルデータタイプの情報がすべて表示されます。テーブル行で選択し、下部パネルにデータを読み込みます。


データタイプの解決を終了したら、OK をクリックして変更を許可します。編集を反映して更新された Reconciler ダイアログに戻ります。

### 10.3.4. Criteria Builder の使用

Transformation Editor の Criteria Builder では、ダイアログのメタオブジェクトを基にして迅速およびグラフィカルにトランスフォーメーションで criteria 句をビルドできます。トランスフォーメーション SQL の既存の基準内にカーソルがある状態で **Criteria Builder** を起動すると、Criteria Builder は Edit モードで開かれます。既存の基準にカーソルがない場合、Criteria Builder は Create モードで開かれ、最初から作成することができます。

ここでは、Criteria Builder を使用して criteria 句をビルドする例を説明します。独自の基準をビルドする場合、必要なロジックで値や定数を組み合わせると強力に複雑な基準をビルドできます。

Criteria Builder の使用方法は次のとおりです。

1. In the **Transformation Editor**, click the **Launch Criteria Builder** button. 
2. **Criteria Builder** が表示されます。

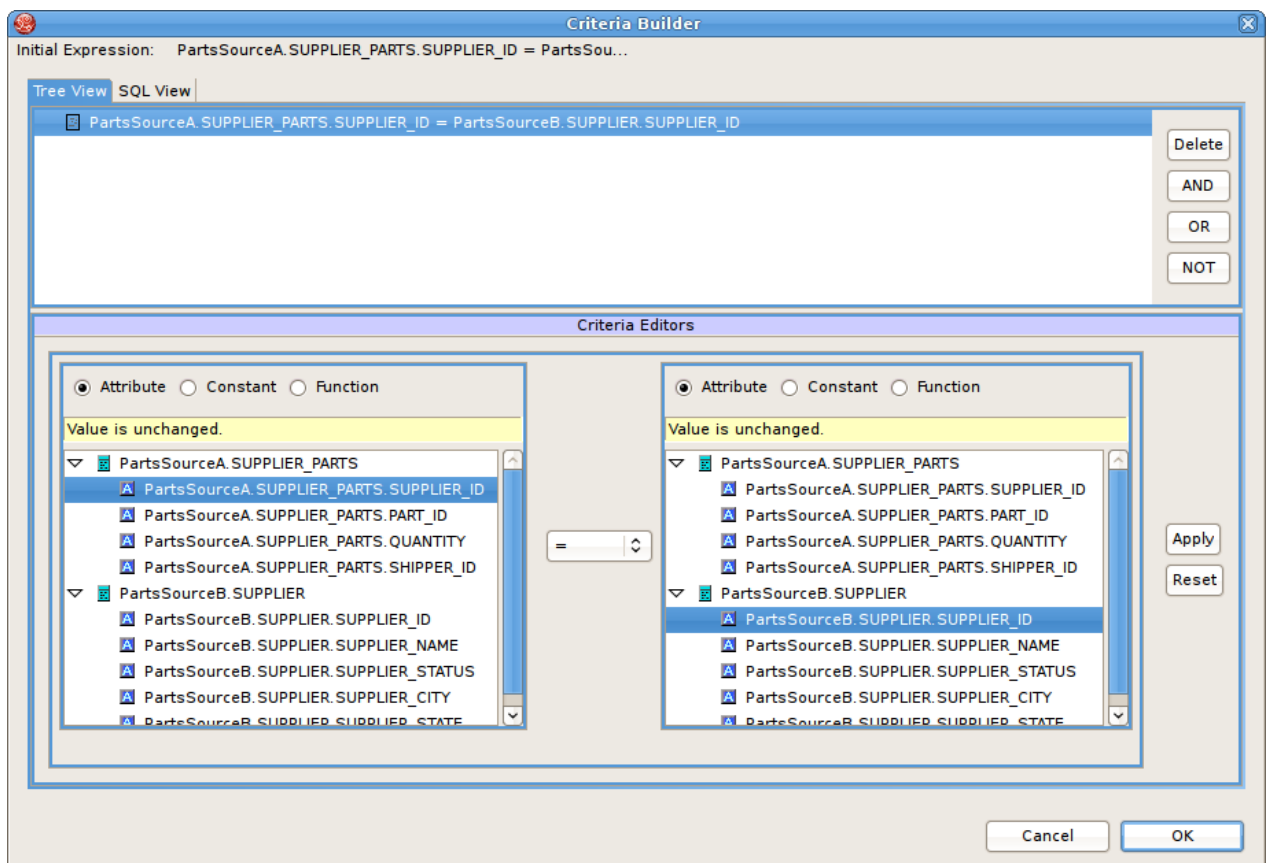


図10.17 String プロパティの編集

上部にある Tree View と SQL View の 2 つのタブには、ビルドした現在の基準の内容が表示されます。

下部の **Criteria Editor** で criteria 句をビルドできます。criteria 句をビルドするには、述語の左側に情報を追加して、比較演算子を選択し、値を右側に追加する必要があります。

3. **Predicate Editor** の両側にあるラジオボタンを使用すると、述語の各側に置く内容のタイプを選択できます。配置したい内容タイプのラジオボタンをクリックします。以下をクリックできません。

- ※ Attribute をクリックすると述語に属性を追加できます。**Attribute** ラジオボタンをクリックすると、Predicate Editor は次のようになります。

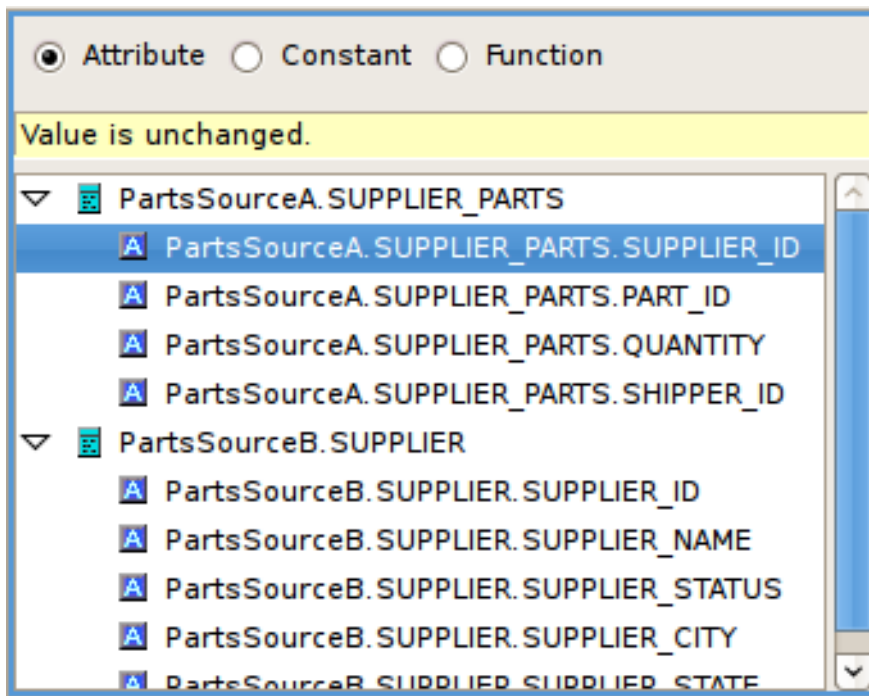


図10.18 Attribute パネル

ツリーから式に追加する属性を選択します。トランスフォーメーションのすべてのソースクラスから属性を選択できます。

- ※ Constant をクリックすると述語にハードワイヤの定数値を追加できます。このラジオボタンをクリックすると、Predicate Editor は次のようになります。

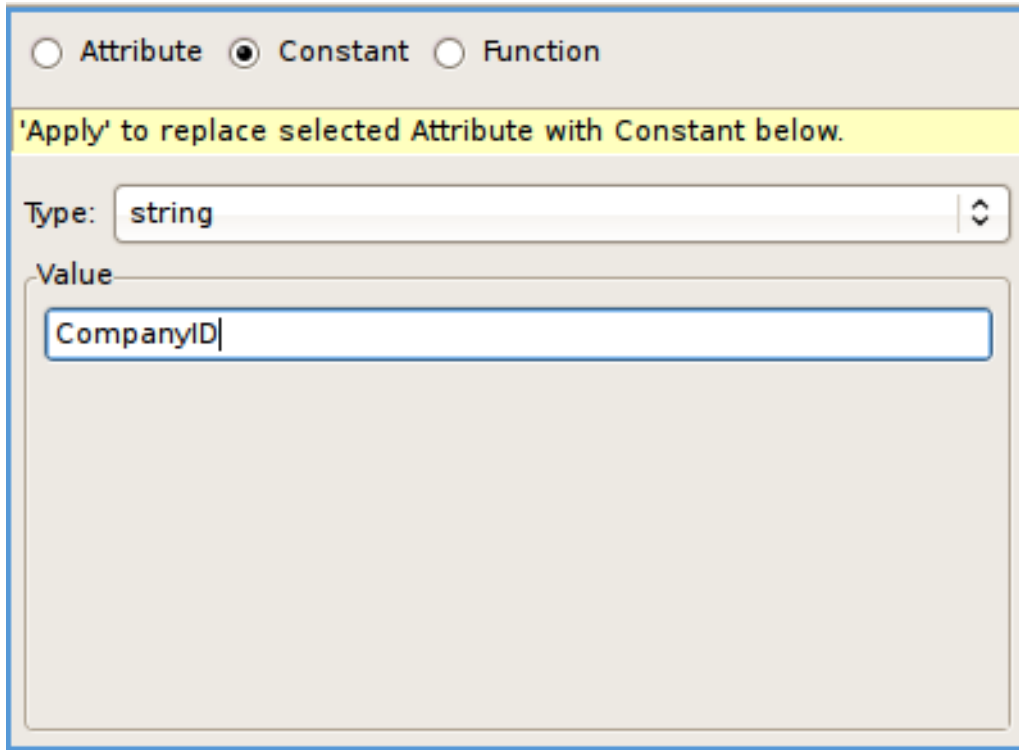


図10.19 Constant パネル

Type ドロップダウンリストからこの定数のデータタイプを選択し、Value 編集ボックスに値を入力します。

※ Function をクリックすると関数を追加できます。

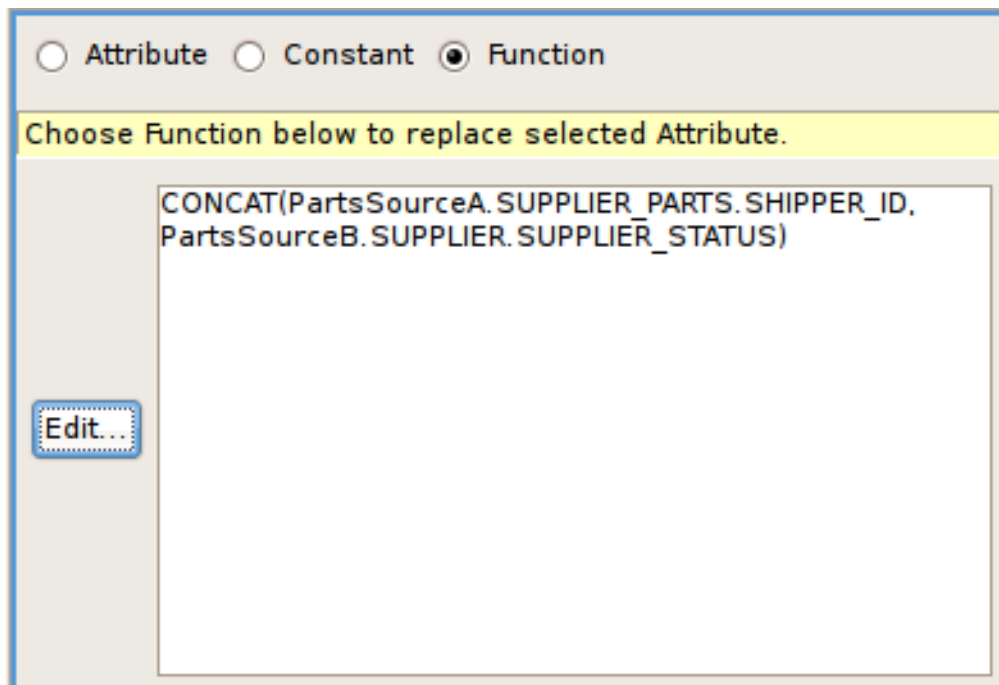


図10.20 Function パネル

SQL 基準の述語で使用する関数の構築に Expression Builder を使用するには、**Edit** ボタンをクリックします。

4. 述語の左側の値を設定し、必要な場合は述語の右側も設定します。述語の右側に値が必要ない場合は、値を入力できません。
5. **Apply** をクリックします。
6. Predicate Editor で左と右の式を両方作成したら、**Apply** をクリックしてダイアログボックス上部のツリービューに基準を追加します。

Criteria ツリーに criteria 句が表示されます。

これに他の基準を連結すると複雑な基準を作成できます。基準を結合するには、Criteria ツリーで基準を選択し、以下のいずれかをクリックします。

- ✧ Delete - 選択した基準を削除します。
- ✧ AND - true にする必要がある新しい基準を作成します。
- ✧ OR - 選択した基準の代わりに true にすることができる新しい基準を作成します。
- ✧ NOT - 否定的な基準を作成します。

作成した基準に他の基準を結合するには、Expression Editor および Predicate Editor パネルを使用して式を同様にビルドします。AND および OR ボタンを使用すると、複雑なネストされた基準を作成できます。

基準を作成し終わったら、**OK** をクリックしてトランスフォーメーションに追加します。

### 10.3.5. Expression Builder の使用

Transformation Editor の Expression Builder では、トランスフォーメーションの式を迅速かつグラフィカルにビルドできます。Expression Builder では以下を作成できます。


- ✧ 属性 (属性を選択して作成)。
- ✧ 定数 (データタイプおよび値を選択して作成)。
- ✧ 標準の Teiid Designer SQL 関数および企業のカスタムユーザー定義の関数からの関数。Expression Builder を起動する前に関数を選択すると、Expression Builder を使用して選択した関数を編集できます。それ以外の場合は、新しい関数を最初から作成できます。



#### 注記

Expression Builder を介して利用できる関数は Teiid Reference Guide に記載されています。

Expression Builder の使用方法は次のとおりです。

1. **Transformation Editor** で関数を挿入する場所をクリックします。
2. Click the **Expression Builder** button.  The **SQL Expression Builder** displays.

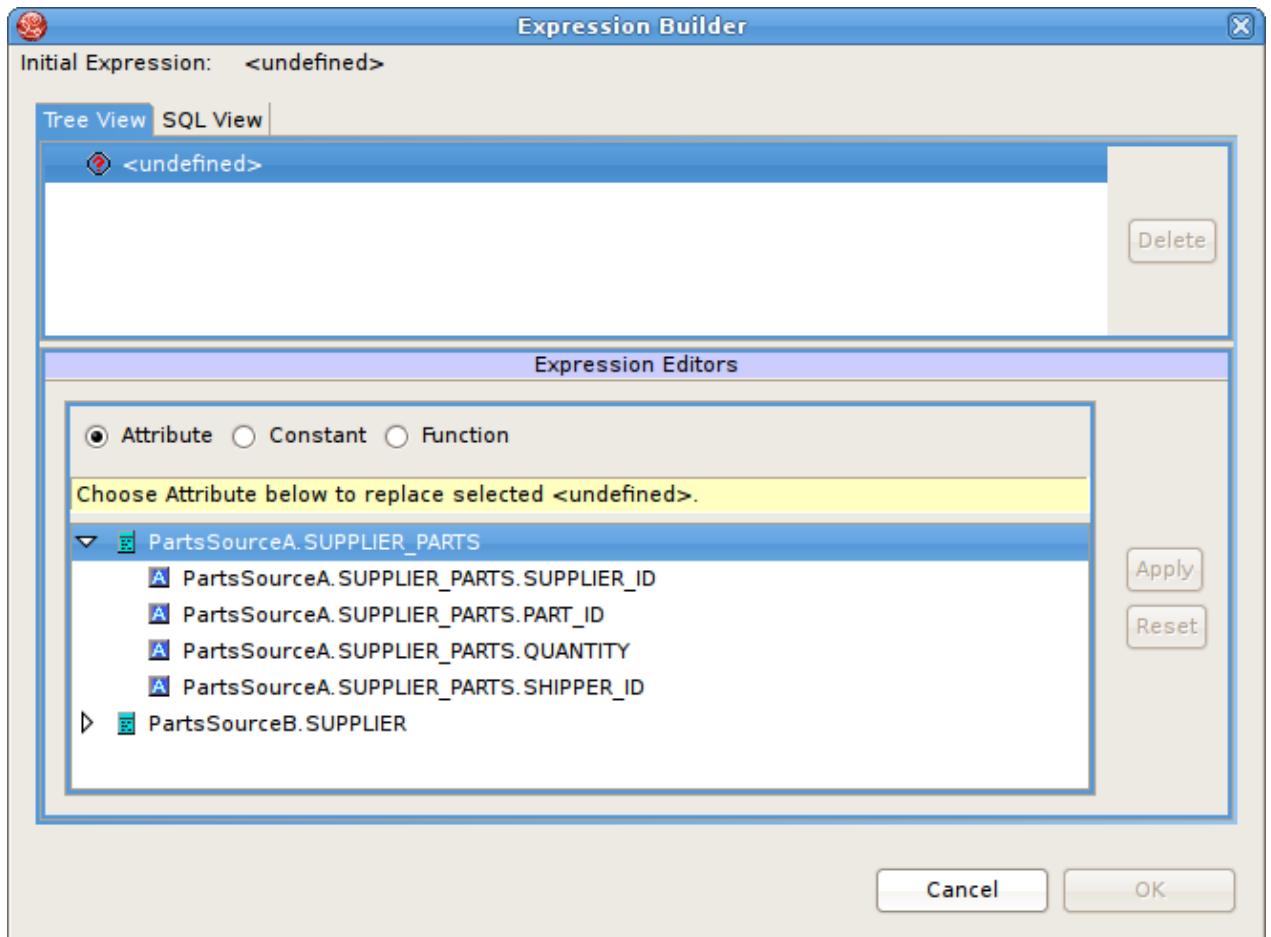


図10.21 Expression Builder

上部にある **Tree View** および **SQL View** の2つのタブには、ビルドした式の現在の内容が表示されます。式をビルドするには、ビルドする式のタイプを指定し、データを入力する必要があります。ほとんどの場合で、Expression Builder を使用して複雑な式を構築します。

3. **Function** ラジオボタンをクリックして関数を追加します。



#### 注記

**Attribute** または **Constant** ラジオボタンを使用すると、定数および属性を式として追加できます。しかし、Expression Editor は関数を扱う場合に最も便利です。

4. **Expression Editor** は関数エディターを表示します。

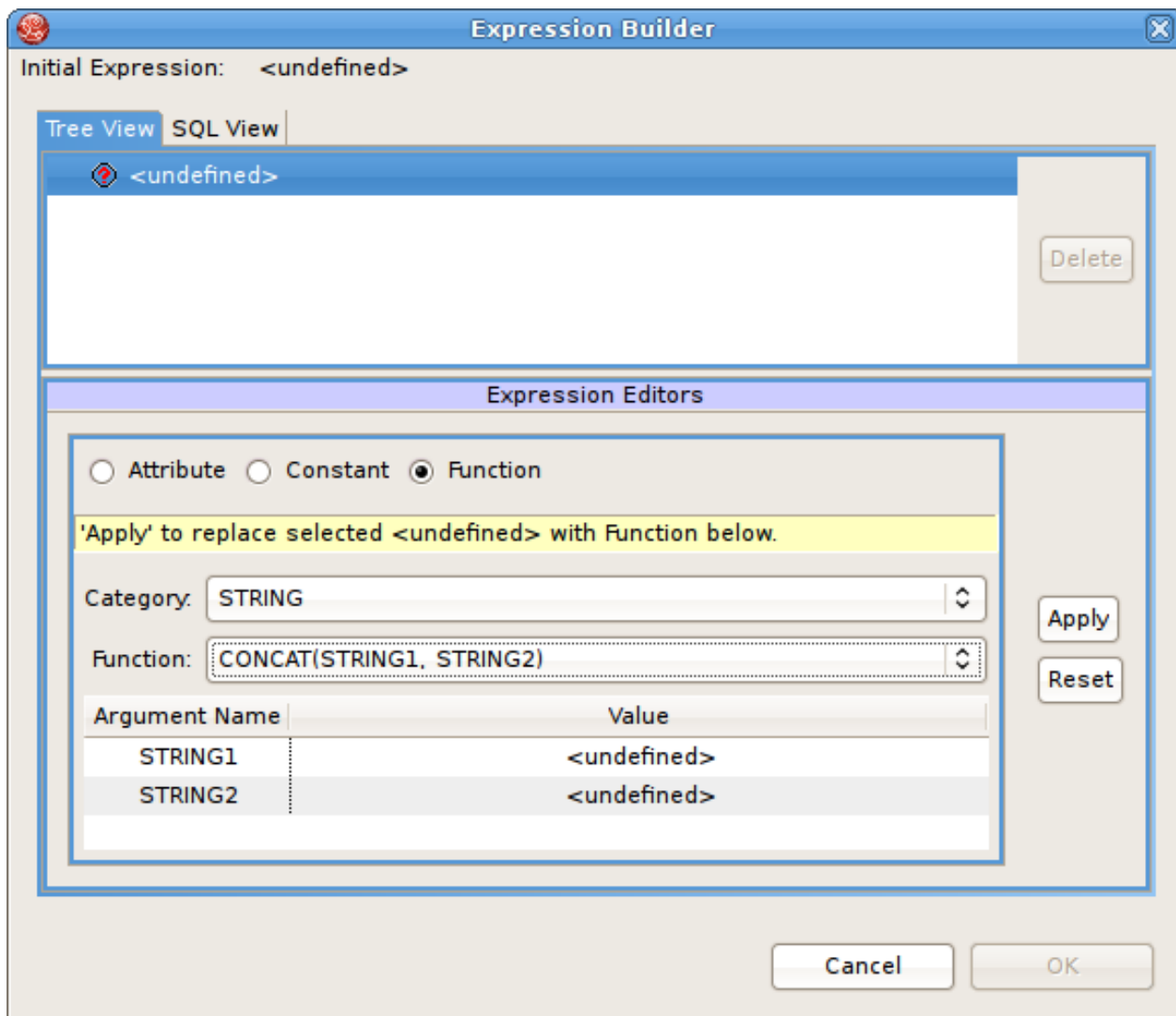


図10.22 関数パネルの選択

**Category** ドロップダウンリストで追加する関数のタイプを選択します。デフォルトでは、**Teiid Designer** は以下のカテゴリーを提供します。

- \* Conversion - データタイプを他のデータタイプに変換する関数
- \* Datetime - 日付または時間の情報を処理する関数
- \* Miscellaneous - その他の関数
- \* Numeric - 数学的関数およびその他の数値関数
- \* String - 文字列操作関数



追加のカテゴリーは、サイトによって作成されたユーザー定義の関数が含まれるものを表します。

5. **Function** ドロップダウンリストで関数を選択します。ドロップダウンリスト下のテーブルにこの関数に必要な引数の数が表示されます。



6. **Apply** をクリックします。
7. 関数がツリーの上部に表示されます。この関数に設定する必要がある各引数に対してサブノードが表示されます。

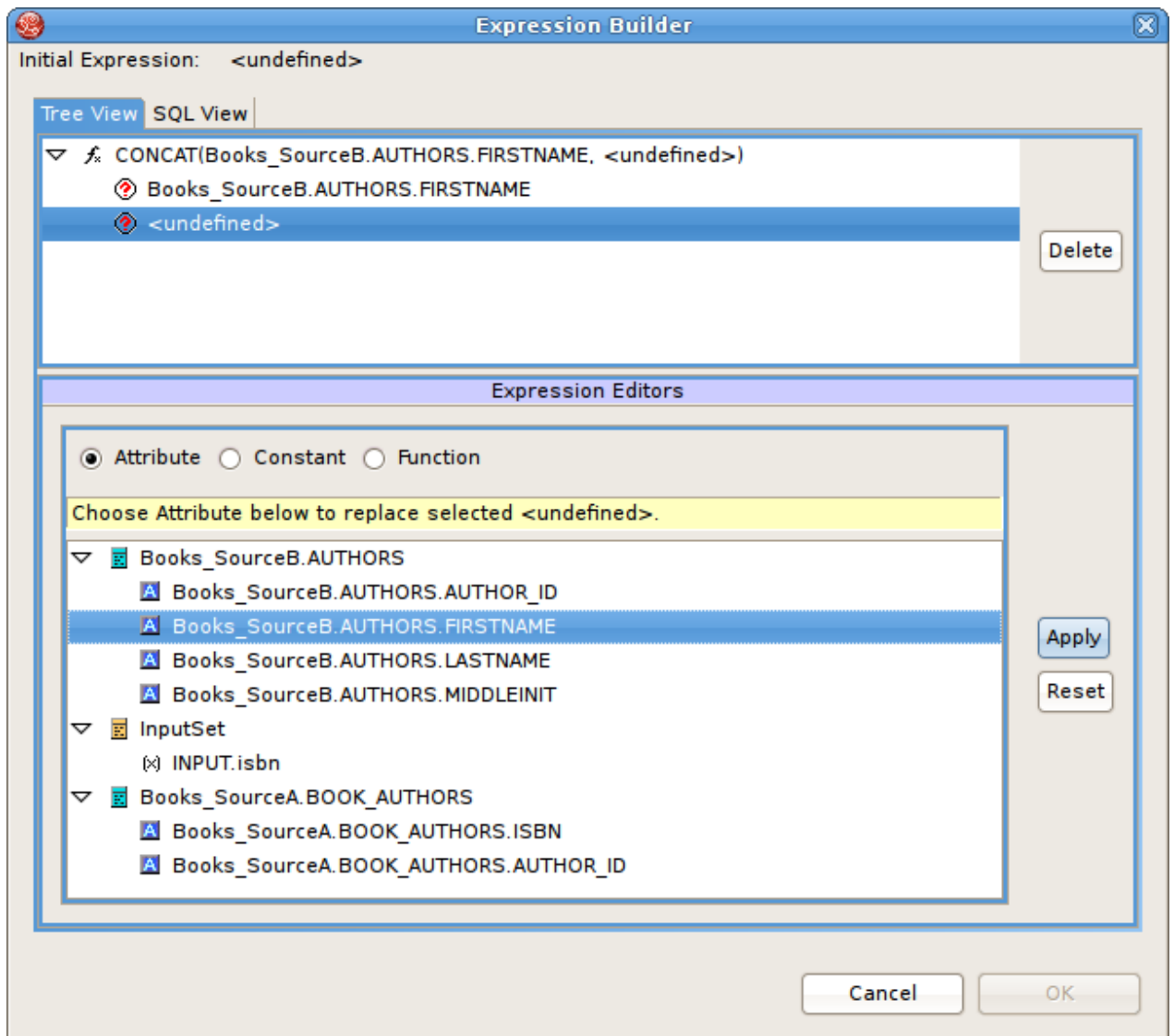


図10.23 新しい空白の関数の作成

ツリーの各サブノードに属性または定数値を設定し、この関数に必要な引数を指定する必要があります。Function エディターを使用してツリーの別の関数をネストすることもできます。

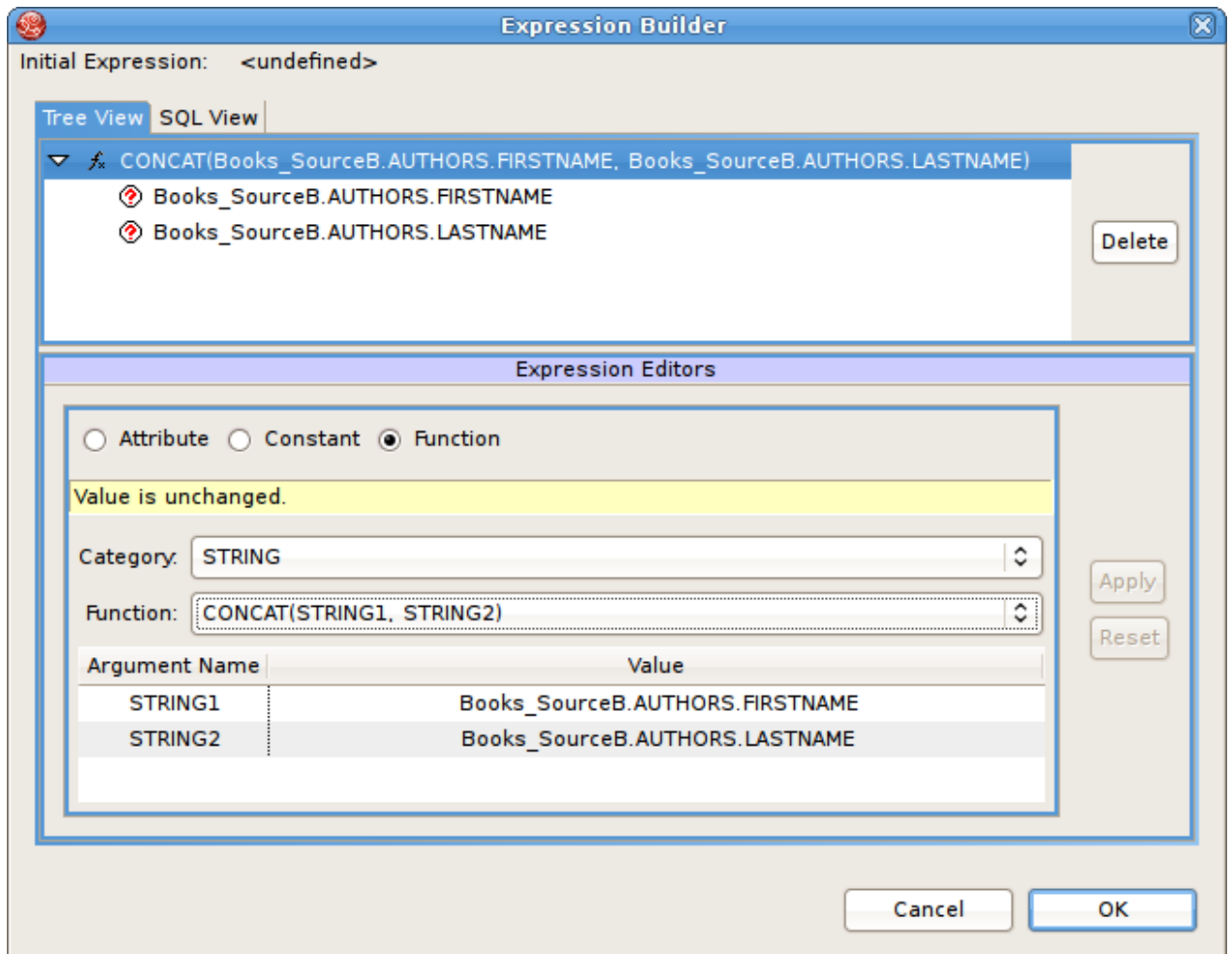


図10.24 ネストされた関数の例

8. ツリーの各サブノードをクリックし、ダイアログボックスの下にあるエディターを使用して属性、定数、または関数値を適用します。
9. 以下のようにすべてのノードに値を追加したら、**OK** をクリックしてこの式をクエリーに追加するか、**Cancel** をクリックして式を挿入せずにダイアログボックスを閉じます。

ツリーのすべてのノードに値が追加されていないと、**OK** ボタンは有効になりません。

引数とその引数の関数を選択し、式内に関数をネストすることもできます。ネストされた関数は、ルート関数の下にあるツリーに表示され、その引数も表示されます。Expression Builder とネストされた関数を使用すると、クエリートランスフォーメーション内で複雑な論理を作成できます。

## 10.4. Input Set Editor (XML)

入力セット (Input Set) は親マッピングクラスからの属性が含まれる特別なクラスを表します。XML ドキュメントモデルのマッピングクラスを作成する場合、**Teiid Designer** によってドキュメントメタオブジェクトの最も高い位置にあるノード下のマッピングクラスの XML トランスフォーメーションすべてに入力セットが自動的に追加されます。

入力セットは **Teiid Designer Server** を使用した情報の統合で使用すると特に便利です。入力セットより、XML ドキュメントの階層で高い位置にあるマッピングクラスの XML トランスフォーメーションによって生成されたデータの行にアクセスできます。子マッピングクラスの XML トランスフォーメーションクエリーの基準内で、データ行からの個別の列である入力セット属性を使用できます。

XML トランスフォーメーションクエリーの SELECT 部分内で入力セット属性を使用することはできません。

入力セットを使用するには、Input Set Editor を使用して親クラスから属性をバインドする必要があります。

入力セットを作成したら、XML トランスフォーメーションダイアグラムのクエリーのソースマテリアルとして入力セット内で属性を使用できます。

入力セットは、ネストされたマッピングクラス間のデータフローの有効化のみを行います。データのアクセスに **Teiid Designer Server** を使用する場合、アプリケーションは入力セットのクエリーを直接実行することはできません。入力セットは、属する XML トランスフォーメーションダイアグラムでのみ表示されます。入力セットは Model Explorer ビューでは表示されず、普通のクラス (他のトランスフォーメーションのソースクラスなど) として使用することはできません。

**Input Set Editor** を開くには、**Mapping Transformation Diagram** の入力セットをダブルクリックするか、ダイアグラムの **Input Set** にある Edit ボタンをクリックします。

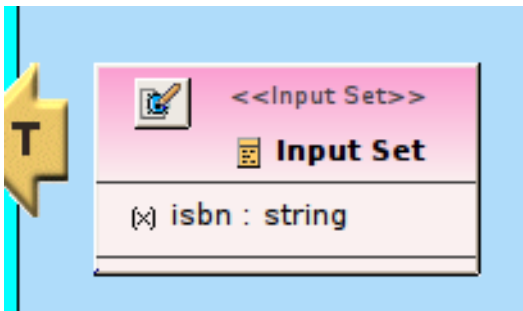


図10.25 Edit Input Set ボタン

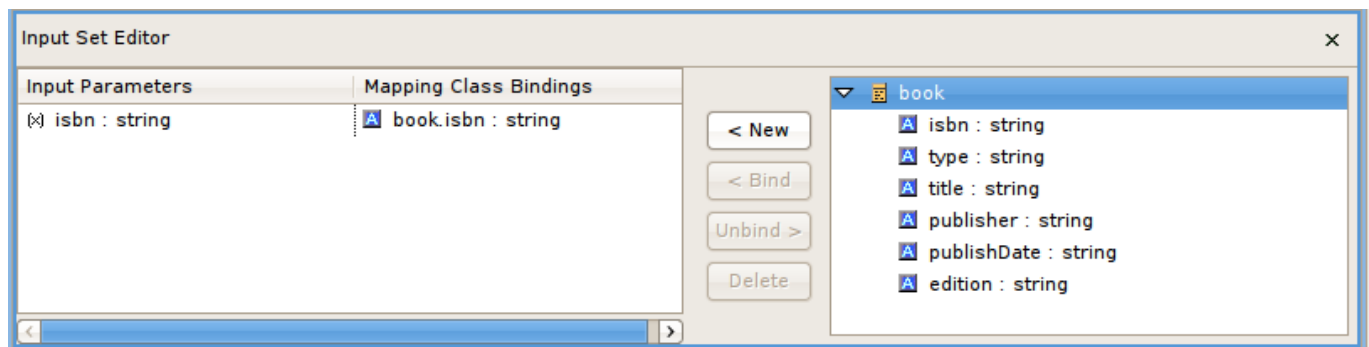


図10.26 Input Set Editor パネル

The Input Parameters テーブルには入力セット内のマッピング属性のリストと、入力セットマッピング属性にバインドされたマッピング属性が含まれます。右側のツリーには親マッピングクラスと各親マッピングクラスから利用できる属性が表示されます。

Input Set Editor を使用すると以下を実行できます。

- ※ 親マッピングクラスから入力セットへマッピング属性を追加できます。右側のツリーで属性を作成するシンボルを選択し、**New** をクリックします。Input Parameters および Mapping Class Bindings テーブルに項目が表示されます。
- ※ 入力セットからマッピング属性を削除できます。Input Parameters および Mapping Class Bindings テーブルで削除する行をクリックし、**Delete** をクリックします。**Teiid Designer** によってテーブルからこの行が削除され、入力セットからこのマッピング属性が削除されます。
- ※ 入力パラメーターをバインドおよびバインド解除できます。

必要な入力セット内でマッピング属性を作成したら、マッピングクラストランスフォーメーション内で入力セットパラメーターを使用し、XML ドキュメントへマッピングできるマッピング属性を生成することができます。

## 10.5. Choice Editor (XML)

### 10.5.1. Choice Editor (XML)

XML ドキュメントモデル内では、XML インスタンスドキュメントのその場所に表示することが可能なドキュメントデータ構造すべてが choice コンポジターによって定義されます。**Teiid Designer Server** が仮想 XML ドキュメントを基にランタイムに XML インスタンスドキュメントを入力すると、Choice Editor 内で指定した基準と一致する最初の断片が選択されます。

Choice Editor を表示するには、マッピングダイアグラムの XML ドキュメントツリービューで choice ノードを右クリックし、右クリックで表示されたポップアップメニューの **Edit** をクリックします。

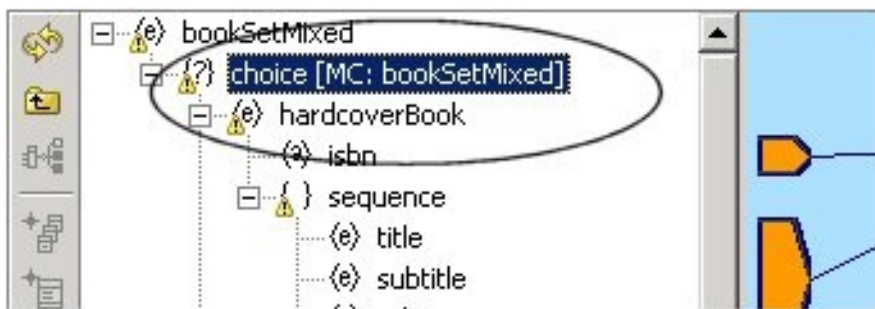


図10.27 Choice Editor を開く

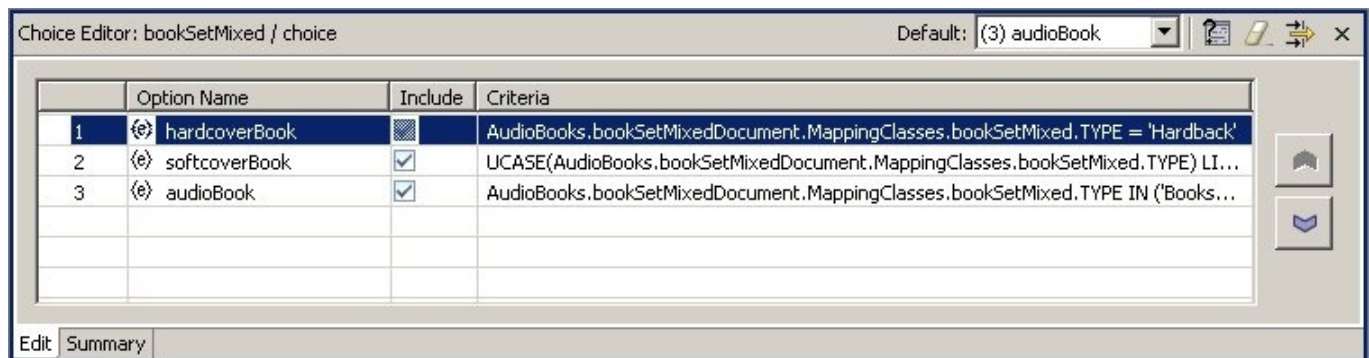


図10.28 Choice Editor

このパネルのテーブルは、ドキュメント断片のトップノードによって表される選択肢の断片オプションを表示します。

以下に示す **Summary** タブは、現在の選択肢基準の SQL に似たバージョンを表示します。

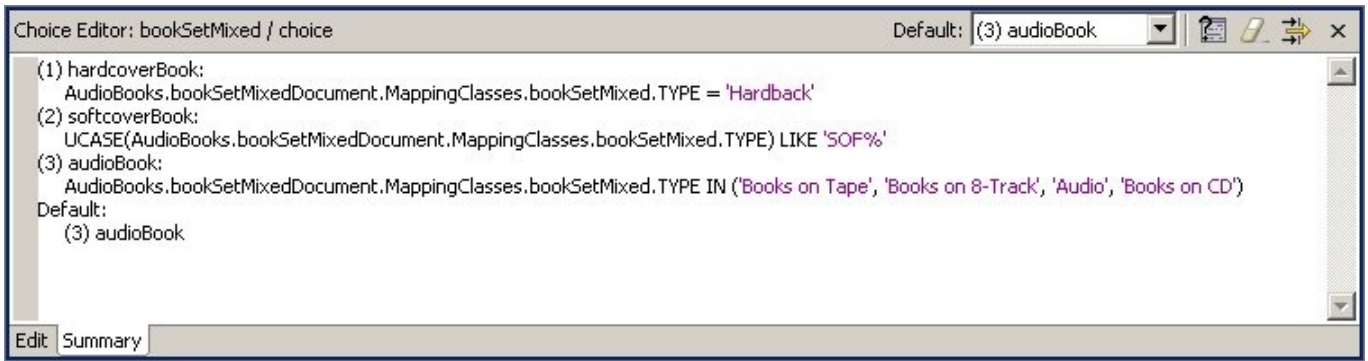


図10.29 選択肢基準の概要

## Choice Editor の使用

以下のいずれかを実行して各選択肢オプションに対処できます。

- ▶ 結果となるドキュメントに挿入する要素を決定するため、適用する **Teiid Designer Server** の基準ステートメントを指定します。
- ▶ ドキュメントにオプションの断片を含めるか、除外します。
- ▶ 要素の基準テストの順番を設定します。
- ▶ 設定した基準と一致しない場合に発生するデフォルトのアクションを設定します。

### 10.5.2. 断片の除外

XML ドキュメントモデルを基にした XML スキーマは選択肢に使用できるオプションの性質を決定します。その他の外部ソース (ビジネスパートナーなど) と共有するスキーマには、XML ファイル内に含めたくない情報が含まれていることがあります。

たとえば、Sample Financial Services がパートナーである Example Mutual Insurance、Illustrative Brokerage、および FinancialPartners.com と XML スキーマを共有するとします。お互いにやり取りする可能性がある情報がすべて含まれるようにするため、パートナーは幅広いスキーマを作成します。そのため、顧客情報の XML ドキュメントには、すべての企業が提供する全製品のリストを基に choice コンポジターが含まれる可能性があります。

しかし、Sample Financial Services はクレジットカードを提供しません。XML ドキュメントにはクレジットカード情報がないため、**Teiid Designer Server** が作成する XML ドキュメントからこのような要素を削除することができます。

**Choice Editor** のテーブルには **Includes** 列が含まれています。デフォルトでは、スキーマによって指定されるすべての要素が含まれます。この仮想 XML ドキュメントのメタデータモデルによって生成される XML ドキュメント内に含めたくない要素の横にあるチェックマークをクリックして削除することができます。チェックマークを削除しても XML ドキュメントモデルからその要素が削除されるわけではなく、**Teiid Designer Server** に選択肢としてこの要素を使用しないように指示するだけです。

除外された要素の基準を編集することはできません。しかし、作成した基準のオプションを除外すると、そのオプションを今後含めたい場合に **Teiid Designer** は基準を保持します。

### 10.5.3. 選択肢基準の編集

choice 要素の基準を編集するには、以下を行います。

1. **Choice Editor** パネルのテーブルで、編集する要素を選択します。

2. **Edit Criteria** ボタンをクリックし、**Criteria Builder** ダイアログを表示します。
3. **Criteria Builder** を使用し、XML インスタンスドキュメントでこのオプションを選択するかどうかを判断するために **Teiid Designer Server** がテストする条件を作成します。
4. **OK** をクリックします。設定した基準は Table および Summary タブの両方に表示されます。

オプションを除外するよう選択した場合や、デフォルトオプションとして指定した場合を除き、ドキュメントの各オプションに基準を設定する必要があります。



#### 10.5.4. choice 要素の順序設定

choice 要素の基準を編集するには、以下を行います。

**Teiid Designer Server** は表示される順で選択肢基準を評価します。1 つの選択肢基準に一致すると **Teiid Designer Server** によってそのオプションで XML インスタンスドキュメントが入力されます。**Teiid Designer Server** はすべてのオプションのすべての基準をテストしない可能性があるため、順序は大変重要になります。

そのため、選択肢基準内でオプションが表示される順序は、多くの場合で最終的に XML インスタンスドキュメントに含まれる情報を決定します。選択肢内のオプションリストの順序を変更し、**Teiid Designer Server** が基準をテストする順序を設定できます。



To set this order, select an element in the table and use the  or  button to move it into a new position in the table. The new order displays both in the table and in the Choice Criteria box and reorders the XML document as well.

#### 10.5.5. デフォルトの Choice アクションの設定

設定した基準のすべてが true に評価されなかった場合、**Teiid Designer Server** はデフォルトのアクションを実行します。

Choice Editor のツールバーにあるコンボボックスを使用して以下をデフォルトとして設定できます。

- ✧ テーブル内のオプション (ドキュメントから除外したものを除く)。
- ✧ THROW (Teiid Designer Server 例外が発生します)。
- ✧ RECORD (Teiid Designer Server 例外を記録します)。
- ✧ DISCARD (XML インスタンスドキュメントに要素を置きません)。



#### 注記

選択肢基準にはデフォルトのアクションを設定する必要があります。

## 10.6. Recursion Editor (XML)

### 10.6.1. 再帰とは

XML スキーマによっては自己参照する要素やデータタイプが含まれるデータ構造を定義するものがあります。XML ドキュメントの生成時、このようなデータ構造はネストされたタグが永遠に繰り返される可能性があります。このような自立的なネスティングパターンを再帰と呼びます。

XML スキーマから仮想ドキュメントを生成するとき、**Teiid Designer** は XML スキーマの再帰的なデータ構造を検出し、再帰的なネスティングパターンが 2 巡した後に停止します。ドキュメントのマッピング時、1 および 2 巡目の目的は異なります。

- ※ 1 巡目は再帰のエントリー条件として考えることができます。このノードにあるマッピングクラスは、普通のマッピングトランスフォーメーションを定義します。
- ※ 2 巡目は、条件が合うまで繰り返し実行されるため Teiid Designer Server によって生成されるドキュメントインスタンスを停止するマッピングトランスフォーメーションを定義します。ドキュメントモデルのこの断片は再帰的な断片と呼ばれます。この断片のマッピングトランスフォーメーションは、1 巡目のマッピングクラス属性にアクセスでき、再帰を停止する条件を指定する機会があること以外は 1 巡目と同じです。

以下に示したように、ダイアグラムオブジェクトの左上隅にある矢印ボタンをループすると、2 番目の再帰的なドキュメント断片に位置するマッピングクラスを認識できます。

再帰が含まれる XML スキーマモデルを基に仮想ドキュメントをモデル化する場合、ネストされた断片を再帰として扱うかどうかを選択できます。データソースからのデータアクセスパターンも再帰的である場合のみ再帰を使用してください (ネストされたドキュメントのデータ内容を生成およびマップするために同じクエリートランスフォーメーションが繰り返し実行される場合)。

デフォルトでは、Teiid Designer Sever で再帰的に実行するために Teiid Designer はドキュメントモデルの再帰的な断片をマーク付けしません。この動作を利用するには、再帰的なマッピングクラスで Recursion Editor を開き、トランスフォーメーションクエリーを再帰的とし、再帰制限プロパティを指定します。

### 10.6.2. Recursion Editor (XML)

**Recursion Editor** では再帰を有効化および制限できます。**Recursion Editor** ボタンは、再帰的なパターンがあるマッピングクラスのみで表示されます。たとえば、Supervisor という要素を含む Employee という要素があり、Supervisor 要素内にネストされた Employee 要素が含まれる場合、ドキュメント内で要素がネストされる回数を制限する必要がある場合があります。

以下の条件を設定すると再帰を制限できます。

- ※ クエリーの固定された結果数。
- ※ SQL ベースの基準制限条件。
- ※ 上記両方の組み合わせ。

To open the Recursion Editor, click the **Recursion Editor** button  on the displayed mapping class.

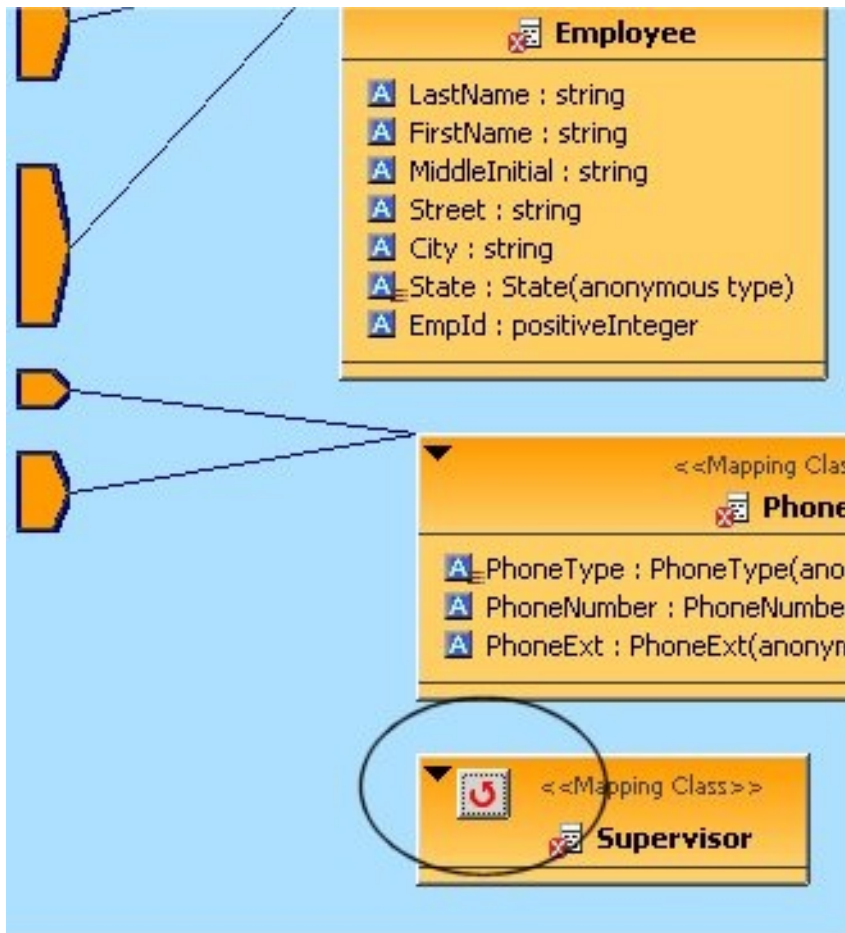


図10.30 Recursion Editor ボタンを開く

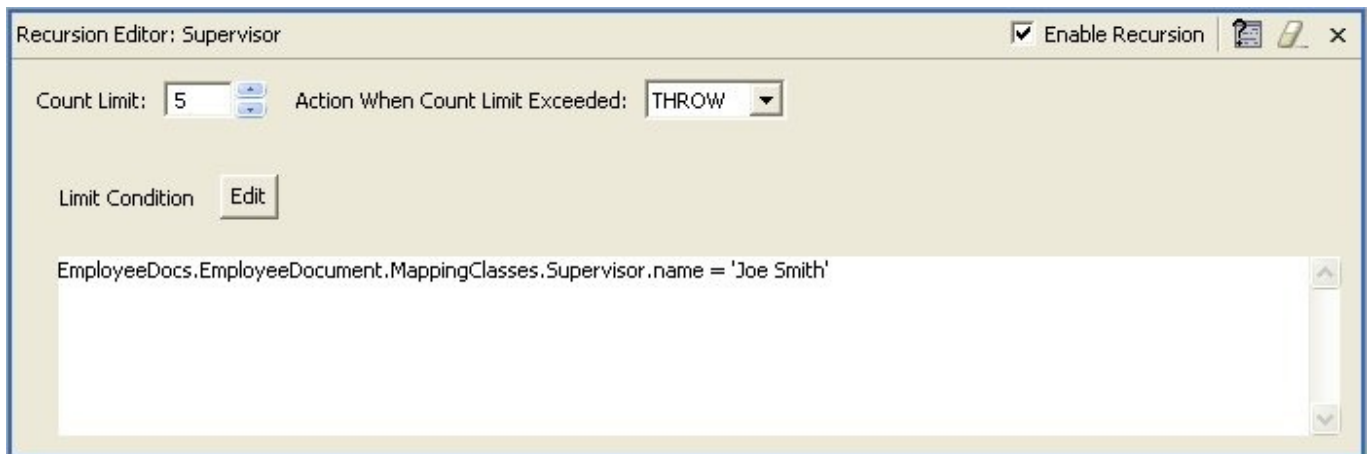


図10.31 Recursion Editor

### 10.6.3. 再帰プロパティの編集

再帰プロパティを編集するには以下を実行します。

1. **Teiid Designer Server** でクエリーを実行し、XML ドキュメント内でネストされたタグを生成するには、Enable Recursion チェックボックスを選択します。
2. **Count Limit** ボックスの横にある矢印をクリックし、クエリーを再帰的に実行する回数を制限します。テキストエリアで **Limit Condition** を設定しないと、クエリーがこの制限に達すると再帰が終了します。 **Teiid Designer Server** によってサポートされる最大数まで制限を設定できます。この制限に関する詳細は、システム管理者に確認してください。



3. **Action When Count Limit Exceeded** ドロップダウンメニューをクリックし、制限条件に達する前にクエリーの結果が回数制限を超えた場合に **Teiid Designer Server** が行うアクションを指定します。
4. **Edit** ボタンをクリックして SQL を起動し、この再帰に対して制限条件を構築します。



### 注記

**Teiid Designer Server** はこのクエリーを再帰的に実行するたびにこの条件を評価します。この criteria 句が false に評価された場合、Count Limit に達していなければ **Teiid Designer Server** によってクエリーが再度実行されます。基準が true と評価されると、**Teiid Designer Server** によって現在のレベルのマッピングが実行され、再帰ループを終了します。

基準を作成すると、**Limit Condition** ボックスに表示されます。

**Teiid Designer Server** によってランタイムに XML ドキュメントが動的に入力されると、ここに入力した再帰の仕様が使用されます。

## 10.7. Operation Editor

Operation Editor を使用すると Web サービス演算トランスフォーメーションの編集が容易になります。Web サービスモデルの編集時、**Operation Editor** という追加のエディターを使用できます。このエディターは以下で構成されます。

- ※ Operations セクション - Web サービスモデル内に含まれるインターフェースおよび演算のツリービューを表示します。
- ※ Input Variable セクション - 入力変数の宣言を編集できます。
- ※ Procedure セクション - プロシーチャーの SQL を編集できます。

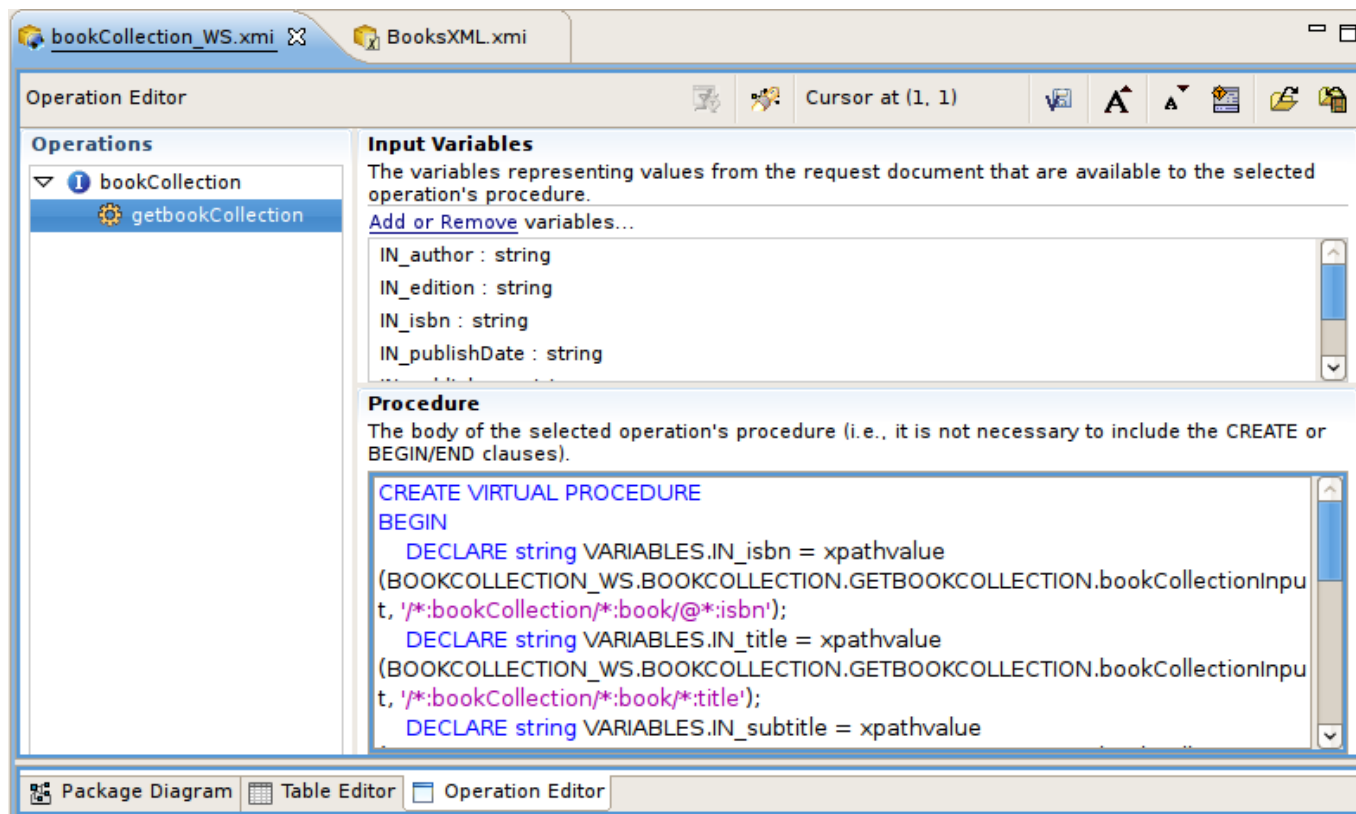


図10.32 Operation Editor

**Operations** セクションには、モデルで現在定義されたインターフェースおよび演算がすべて含まれています。

演算を選択すると、入力パラメーターの内容に関連する変数が **Input Variables** セクションに表示され、そのプロシーチャーのボディ (CREATE VIRTUAL PROCEDURE BEGIN - END キーワード、入力変数宣言、割り当てを除く) が **Procedure** セクションに表示されます。

SQL を貼り付けるとき、CREATE VIRTUAL PROCEDURE BEGIN - END キーワードを含めないでください。演算の入力パラメーターで Content via Element プロパティーが設定されると、入力変数が自動的に生成されます。入力変数は、**Input Variables** セクションの **Add** または **Remove** リンクを使用して編集でき、入力内容内の単一の属性および要素への XPath のみを表す可能性があります。その他の変数宣言および割り当ては **Procedure** セクションに直接入力する必要があります。 **Add** または **Remove** リンクをクリックすると、以下のダイアログが表示されます。

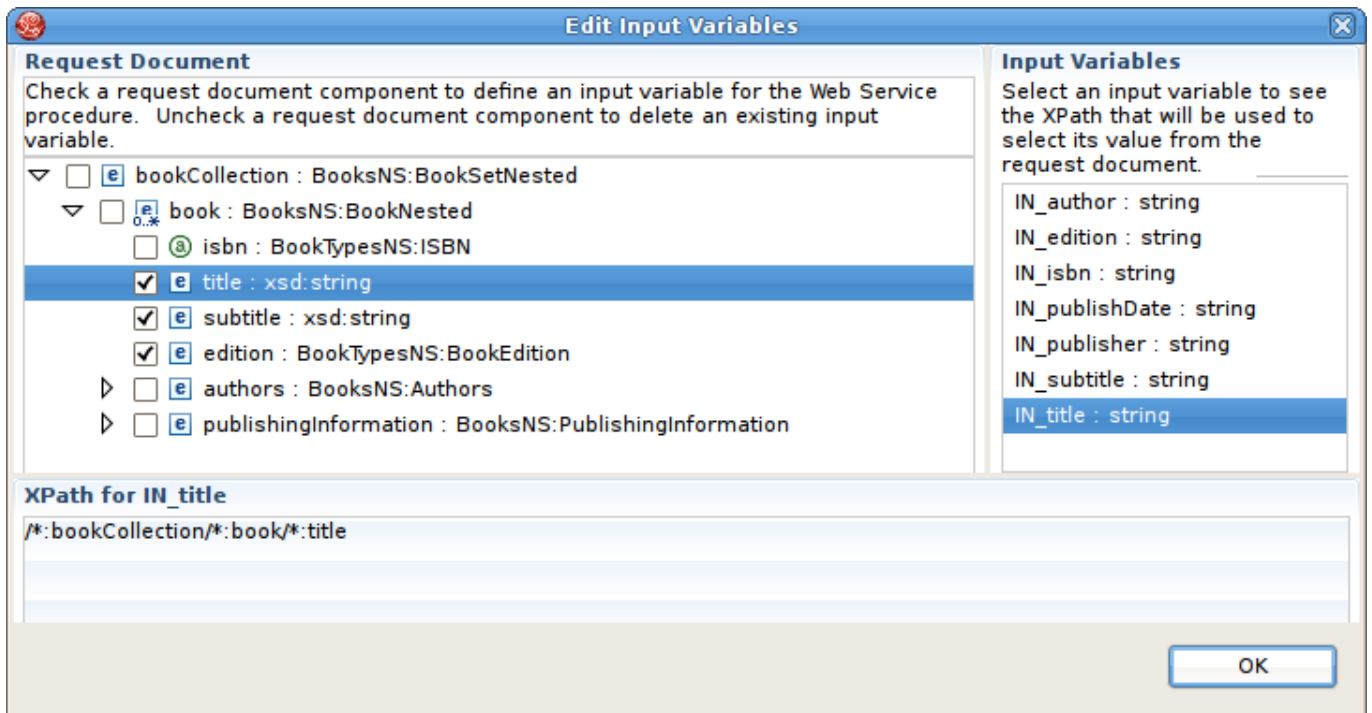


図10.33 Edit Input Variables ダイアログ

## 10.8. モデルオブジェクト拡張の管理

### 10.8.1. モデルオブジェクト拡張の管理

モデルを拡張すると、そのモデルオブジェクトに追加のプロパティが加えられます。一例として、これらの拡張プロパティを使用するとカスタマイズされた Data Services トランスレータにデータを渡すことができます。Teiid Designer モデル拡張フレームワークは以下で構成されます。

- ✦ Model Extension Definitions (MEDs)
- ✦ MED レジストリー - ワークスペースに登録されたすべての MED を把握します。モデルの拡張には登録された MED のみを使用できます。
- ✦ MED エディター

### 10.8.2. 新しい MED の作成

新しい MED を作成するには、**File > New > Other...** と選択し、新しいウィザードダイアログを表示します。Teiid Designer > Teiid Model Extension Defn オプションを選択し、**New Model Extension Definition** ダイアログを表示します。既存プロジェクトまたは MED ファイルのプロジェクトフォルダーの場所を閲覧および選択し、一意なファイル名を指定したら **Finish** をクリックします。



#### 注記

ウィザードが起動したときにプロジェクトがすでに選択される場合は、場所のフィールドが事前入力されます。

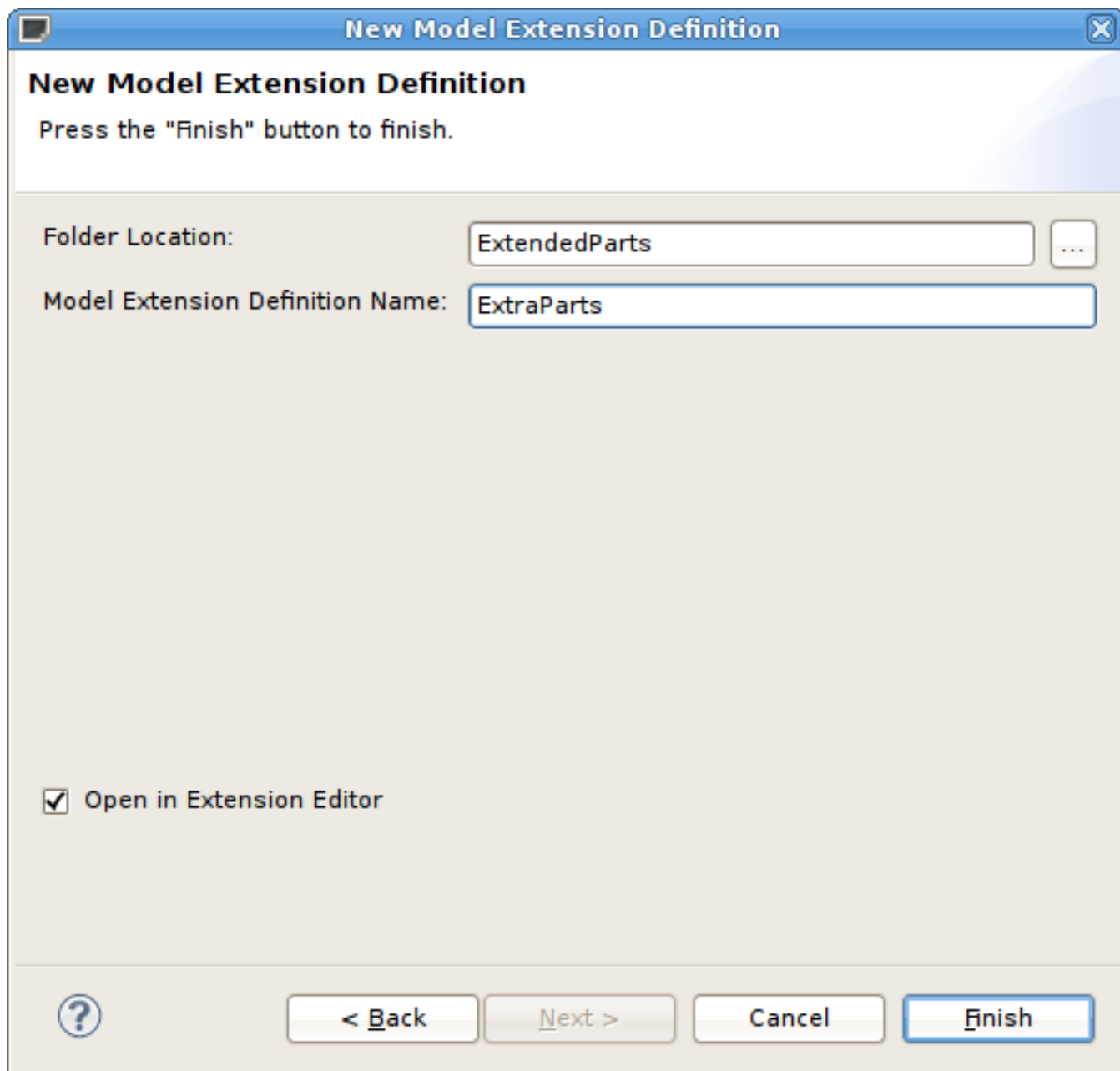


図10.34 MED エディター概要タブ

### 10.8.3. MED の編集

MED ファイルを編集するには、ワークスペースで既存の `.mxd` ファイルを選択し、右クリックで **Open** を選択します。**MED Editor** が開かれ、編集できるようになります。

**Overview** タブでは、名前空間接頭辞、名前空間 URI、拡張するモデルクラス (リレーショナル、Web サービス、XML ドキュメント、および関数)、および説明を指定または変更できます。

基本の MED 情報を入力した後、**Properties** タブに切り替えて、選択したモデルクラスによってサポートされる特定のモデルオブジェクトの拡張プロパティ定義を作成できます。

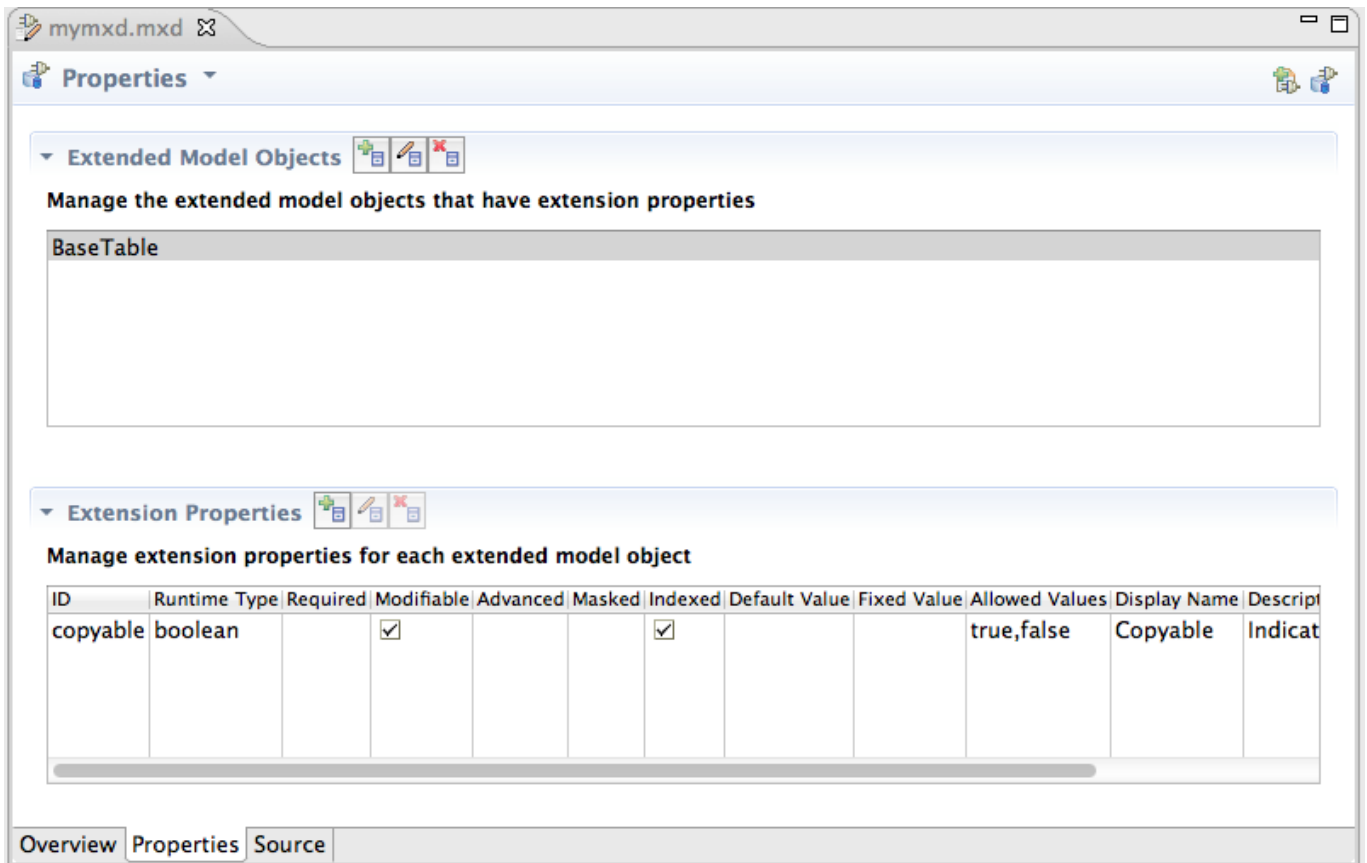


図10.35 MED エディターの Properties タブ

**Add Extended Model Object** ツールバーボタンをクリックし、**Model Object Name** 選択ダイアログを表示します。オブジェクトを選択し、**OK** をクリックします。

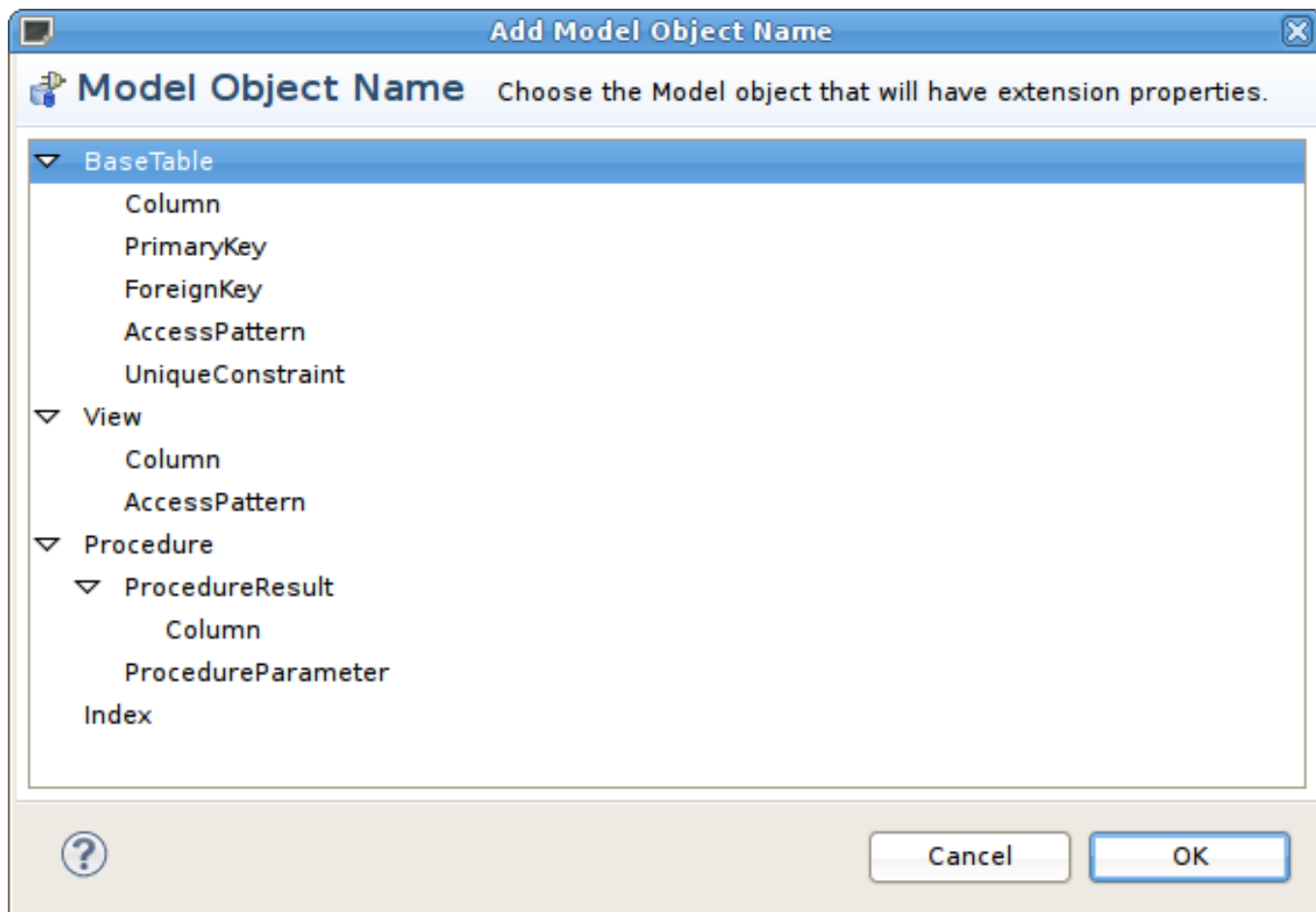


図10.36 Add Model Object Name ダイアログ

次に、**Extended Model Objects** セクションでモデルオブジェクトを選択し、下にある **Extension Properties** セクションのアクションおよびプロパティテーブルを使用して実際の拡張プロパティを追加、削除、または編集します。拡張プロパティの追加または編集アクションを選択すると、一般的なプロパティ、値の定義 (必須の値、マスクされた値、許可される値)、国際化できる表示名および記述値の編集セクションが含まれるダイアログが表示されます。

**Edit Property Definition**

**Property Definition** Enter the property's general information, value information, descriptions, and display names.

**General**  
 Enter general property characteristics  
 Model Object: BaseTable  
 Namespace Prefix: mymodelextension  
 ID: copyable  
 Runtime Type: boolean  
 Should only be modified by advanced users  
 Will be used by Teiid server

**Value Definition**  
 Enter the characteristics of the property value  
 A value is required  
 Mask the property value when shown to user  
 The value must be one of the following:  
 true  
 false  
 Use this initial value:  
 The value can only be this value

**Display Name**

**Description**

Cancel OK

図10.37 Edit Property Definition ダイアログ

#### 10.8.4. MED でのモデルの拡張

モデルのモデルオブジェクトが拡張プロパティを使用できるようにするには、MED をそのモデルに適用する必要があります。特定のモデルに適用された MED を管理するには、モデルを選択し、右クリックして **Modeling > Manage Model Extension Definitions** を選択します。これにより、現在適用されている MED を表示するダイアログが表示され、モデルから MED を追加または削除するボタンが表示されます。モデルから MED を抽出し、コピーが .mxd ファイルとしてローカル保存され、最後に MED レジストリーのバージョンと異なる場合はモデルの MED のバージョンが更新されます。

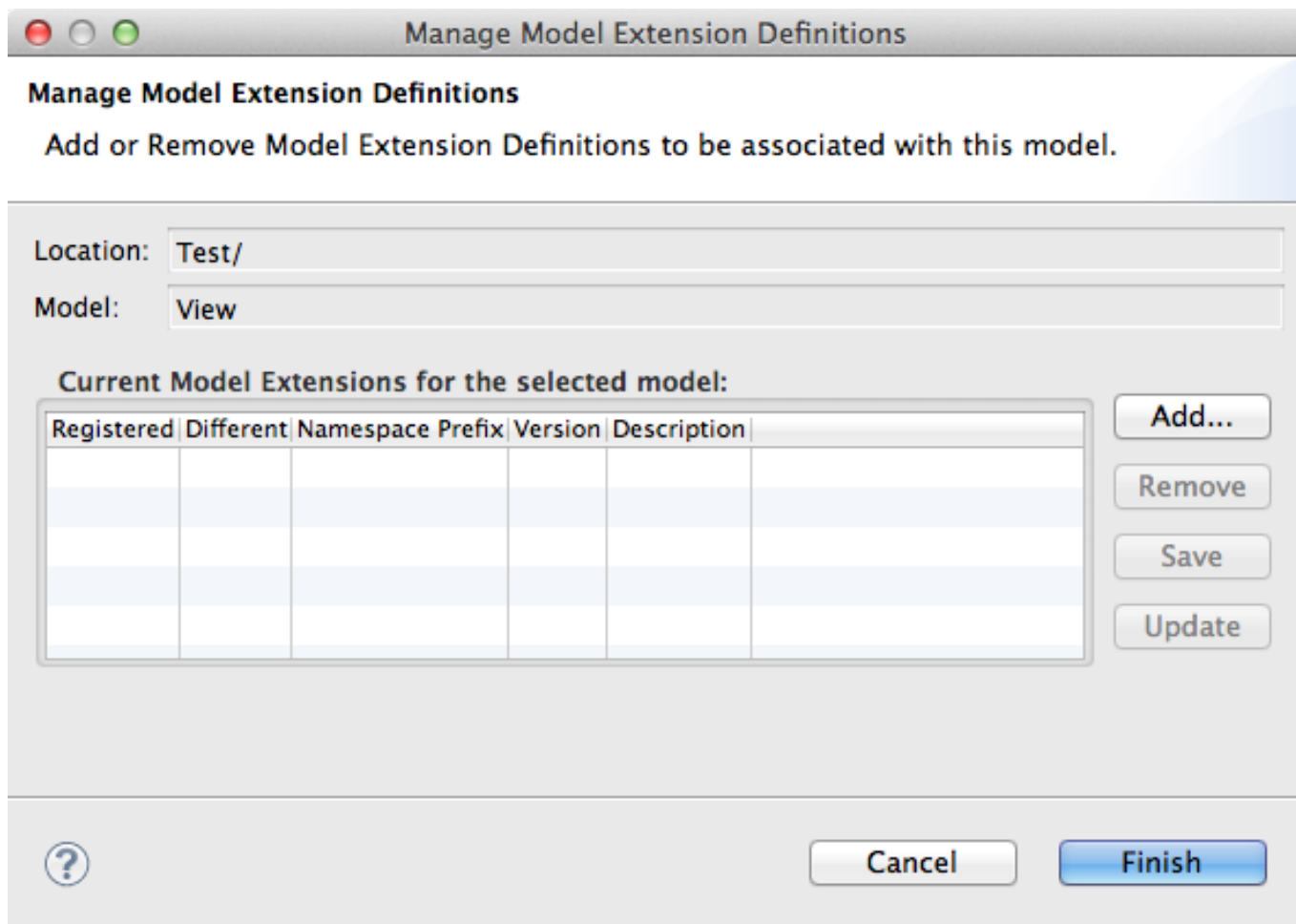


図10.38 Manage Model Extension Definitions ダイアログ

**Add** をクリックして、モデルクラスを基に適用可能な MED のリストを表示します。

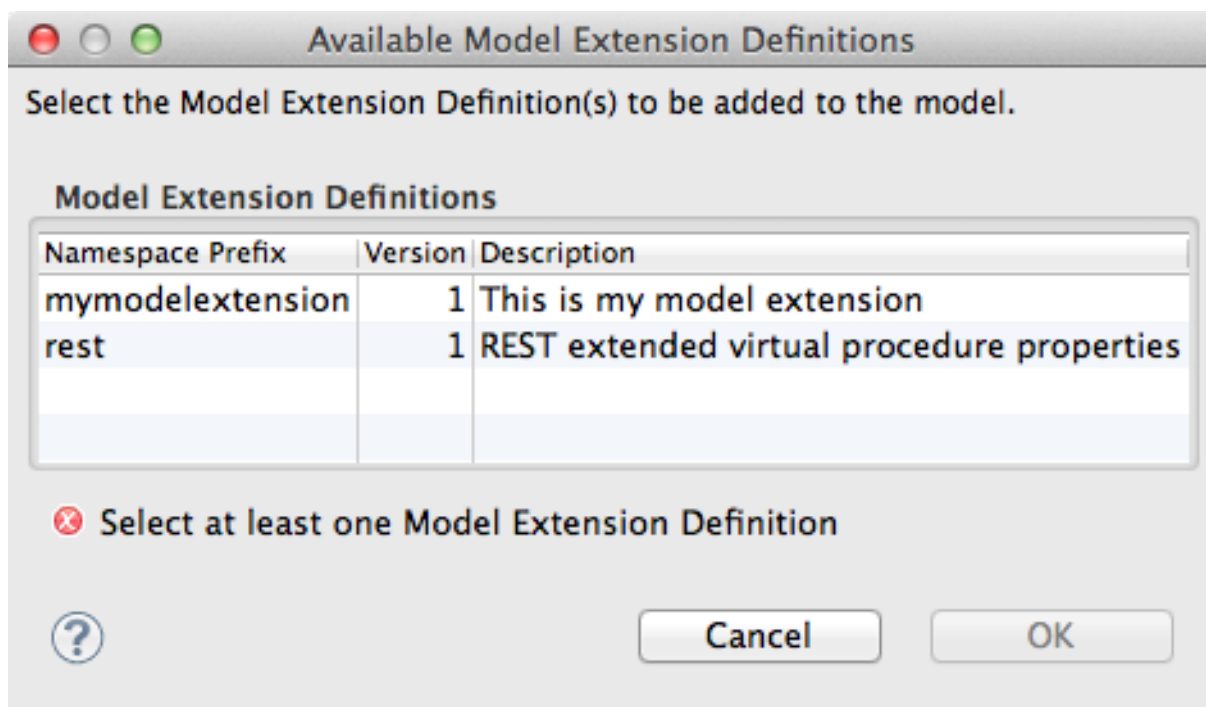


図10.39 Add Model Extension Definitions ダイアログ



## 注記

モデルから MED を追加または削除した後、**Finish** をクリックしてすべての変更を適用します。ダイアログをキャンセルするとすべての変更が破棄され、元のモデルの状態に戻ります。

### 10.8.5. 拡張プロパティ値の設定

拡張プロパティは **Properties View** で拡張モデルオブジェクトが利用できるユーザー定義のプロパティです。すべての拡張プロパティは Extension カテゴリ下で使用でき、接頭辞として MED の名前空間接頭辞が付きます。拡張プロパティの初期値がある場合、MED のプロパティ定義を使用してデフォルト値に設定されます。

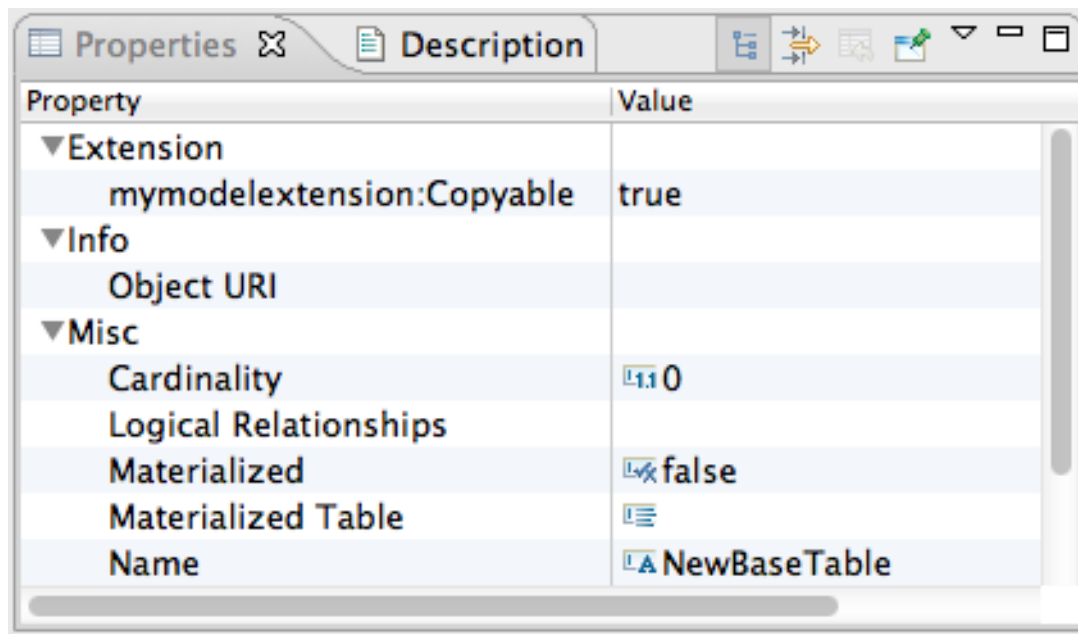


図10.40 拡張モデルオブジェクトの Properties View

## 第11章 ユーザー定義関数

### 11.1. ユーザー定義関数

Teiid Designer では、カスタムまたはユーザー定義関数 (UDF) を使用して Teiid の関数ライブラリーを拡張できます。以下のプロパティーを用いて UDF を定義できます。

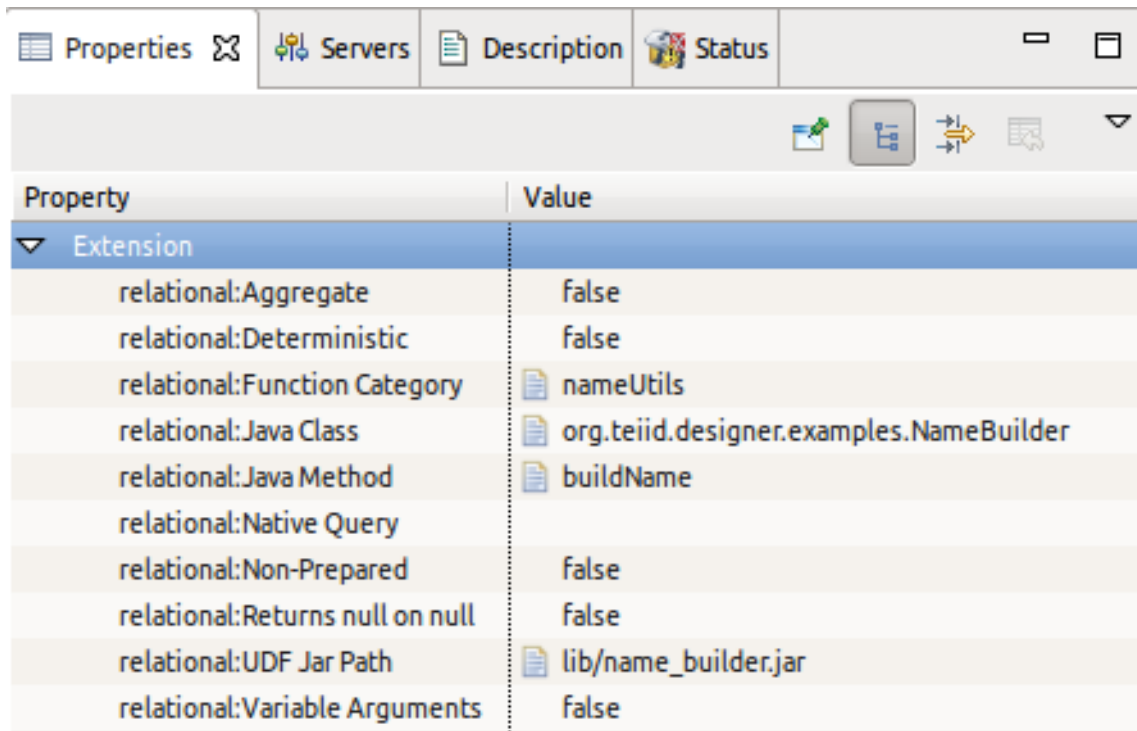
- ✦ 関数名 - 以下を念頭に置いて関数名を作成してください。
  - 既存の Teiid System 関数をオーバーロードできません。
  - 関数名は、引数の数に対してモデルのユーザー定義関数全体で一意でなければなりません。異なる数の引数タイプでは同じ関数名を使用できるため、ユーザー定義関数をオーバーロードできます。
  - 関数名に「.」文字を含めることはできません。
  - 関数名は 225 文字以下でなければなりません。
- ✦ 入力パラメーター - 入力パラメーターはタイプ固有の署名リストを定義します。すべての引数は必要であると見なされます。
- ✦ 戻り値の型 - 戻されるスカラー値の予想される型。
- ✦ プッシュダウン - 予期されるプッシュダウン動作を示します。REQUIRED、NEVER、または ALLOWED のいずれかになります。NEVER または ALLOWED が指定されると、関数の Java 実装を提供する必要があります。REQUIRED が指定されると、ソースのトランスレーターを拡張し、この関数をプッシュダウン関数ライブラリーに追加する必要があります。
- ✦ invocationClass/invokeMethod - UDF がプッシュダウンされなかったときに呼び出すメソッドを示すオプションのプロパティー。
- ✦ 決定論 - メソッドが常に同じ入力パラメーターに対して同じ結果を返すかどうか。デフォルトは false です。パフォーマンスが向上するため、同じ入力に対して同じ値を返す場合は関数を決定論としてマーク付けることが重要になります。

プッシュダウンが必要な関数でも、Teiid が関数を適切に解析および解決できるようにするため、UDF として追加する必要があります。プッシュダウンスカラー関数は通常のユーザー定義関数とは異なり、エンジンでの評価にコードが提供されません。プッシュダウンが必要な関数が適切なソースによって評価できない場合は例外が発生します。

### 11.2. 関数のモデル化

Teiid Designer はユーザー定義関数をモデル化するためのオブジェクトウィザードを提供します。既存のリレーショナルビューモデルを右クリックし、New Child - Procedure をクリックした後、"User Defined Function" のラジオボタンをクリックします。ウィザードではファイルシステムでの関数の jar の場所と、Java クラスおよび Java メソッドを指定するオプションが表示されます。

その他のメソッドを使用して関数を作成する場合は、プロシージャーを選択して **Properties** ビューのプロシージャーの異なるプロシージャーを編集する必要があります。



| Property                        | Value                                   |
|---------------------------------|---|
| Extension                       |   |
| relational:Aggregate            | false                                   |
| relational:Deterministic        | false                                   |
| relational:Function Category    | nameUtils                               |
| relational:Java Class           | org.teiid.designer.examples.NameBuilder |
| relational:Java Method          | buildName                               |
| relational:Native Query         |   |
| relational:Non-Prepared         | false                                   |
| relational>Returns null on null | false                                   |
| relational:UDF Jar Path         | lib/name_builder.jar                    |
| relational:Variable Arguments   | false                                   |

図11.1 VDB UDF Jar ファイル

ユーザー定義関数は、Java クラスの定義されたメソッドを表します。Teiid ランタイムが関数を登録し、トランスフォーメーションで使用できるようにするには、以下をプロパティとして指定する必要があります。

- ※ 関数カテゴリー - ユーザー定義関数のグループ化に対する一意名。このカテゴリーは Expression Builder ウィザードに表示されるため、関数を見つけ選択することができます。
- ※ Java クラス - 関数メソッドが含まれる Java クラスの完全修飾名。
- ※ Java メソッド - Java クラスの関数メソッドの名前。
- ※ UDF Jar パス - プロジェクトの lib フォルダーに存在する UDF が含まれる jar に対するプロジェクトの相対パス。初めて jar を定義し、lib フォルダーに jar がない場合、プロパティエディターでファイルシステムから jar を選択することができます。lib フォルダーに 1 つ以上の jar が存在する場合、エディターでワークスペースまたはファイルシステムのどちらから指定するかを選択できます。

### 11.3. トランスフォーメーションでの関数の使用

関数をモデル化した後、トランスフォーメーション SQL で関数呼び出しを使用できます。これらの関数は Transformation Editor の Expression Builder ウィザードからアクセスできます。

### 11.4. VDB に関数を含める

JBoss Data Virtualization が関数を認識するには、実際のコードがサーバーにデプロイされ、Teiid サブモジュールが利用できるようにする必要があります。Teiid Designer ワークスペースは、関数およびそれらの参照された jar およびクラス情報が含まれるモデルを認識します。ユーザー定義関数が含まれるソースモデルが VDB に追加されると、定義された関数が含まれる jar も VDB に追加され、VDB エディターの UDF Jars タブに表示されます。

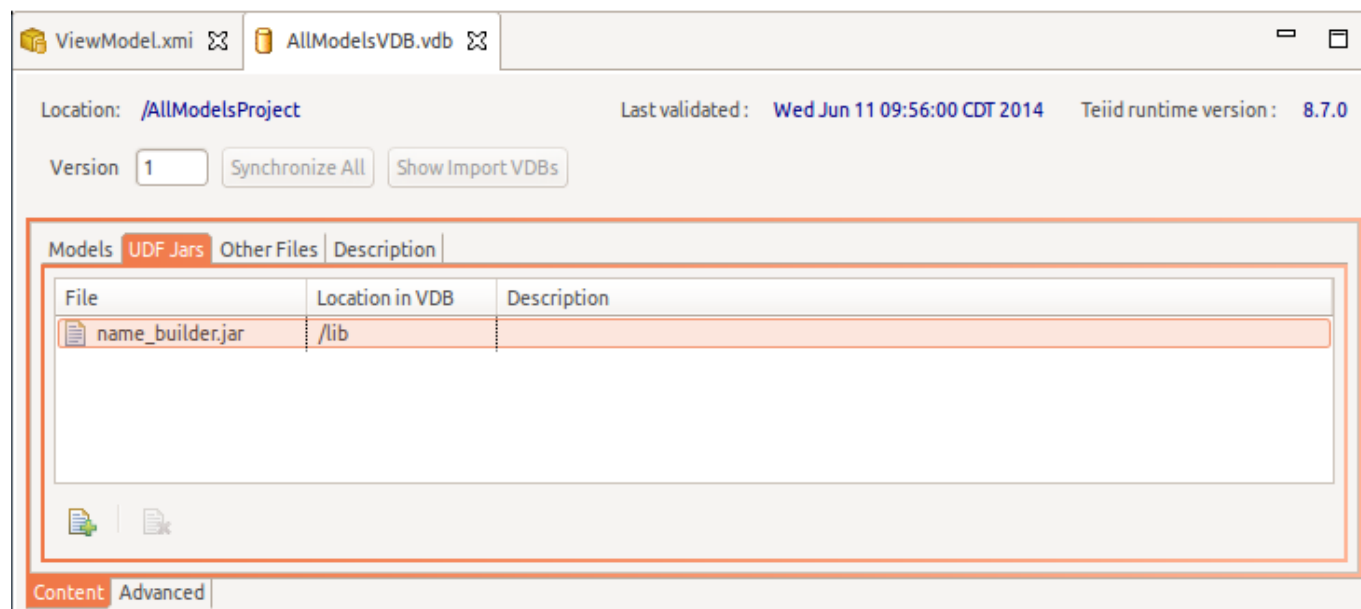


図11.2 VDB UDF Jar ファイル

## 第12章 メタデータ固有のモデリング

### 12.1. リレーショナルソースのモデリング

#### 12.1.1. ソース関数

View トランスフォーメーション内でデータベース関数を使用する機能を向上するため、Source Function アクションとウィザードが Teiid Designer に追加されました。Source Function を使用すると、入力および出力パラメーターを含む関数構造に準拠するソースプロシージャを構築することができます。**New Child > Procedure** とクリックし、**Procedure Type** ダイアログを開きます。

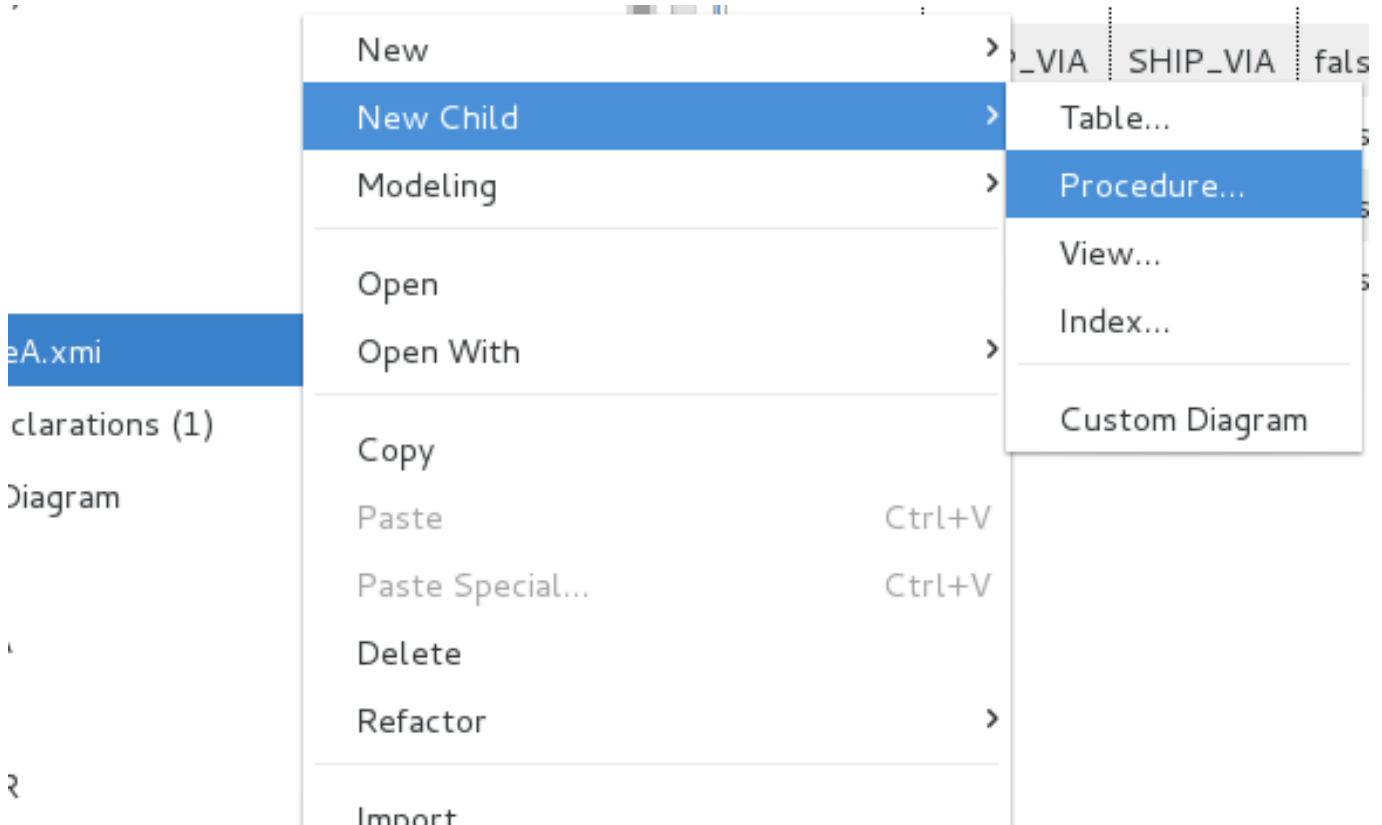


図12.1 新しいソース関数アクション

**Source Function** アクションを選択し、**Create Relational Source Function** ダイアログを開きます。データベース関数名を入力し、データタイプや長さを含む入力パラメーターを定義します。出力パラメーターの情報を指定し、**OK** をクリックします。

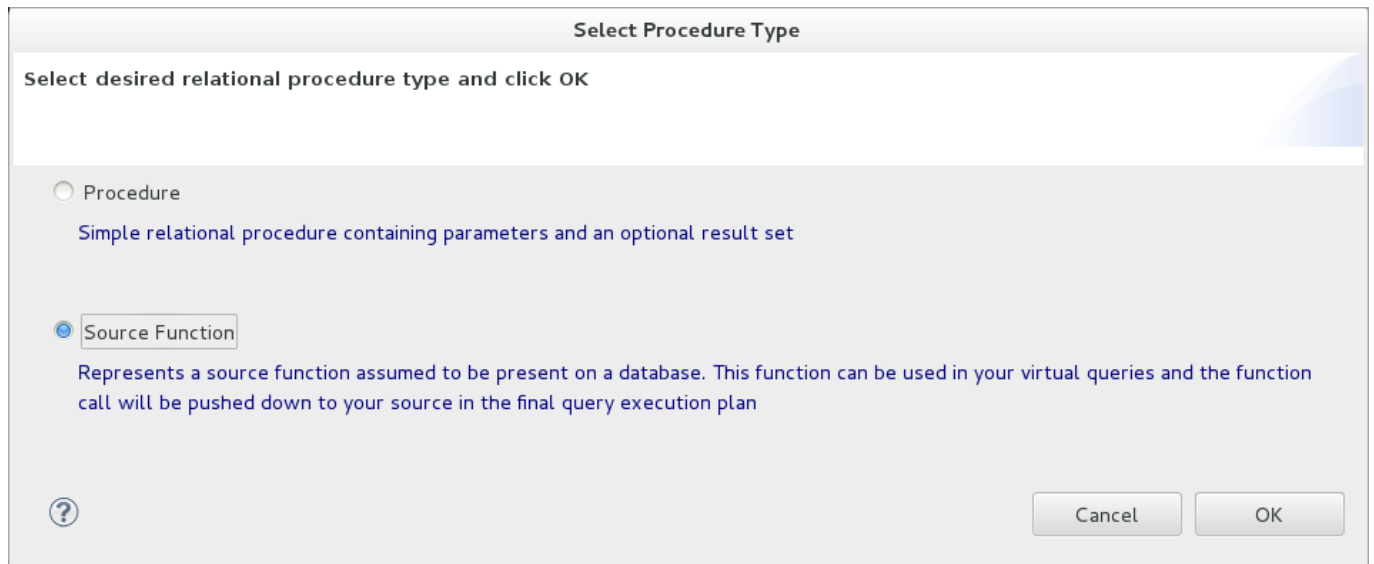


図12.2 新しいソース関数アクション

The resulting source function will be added to your model and will be represented by the  icon.

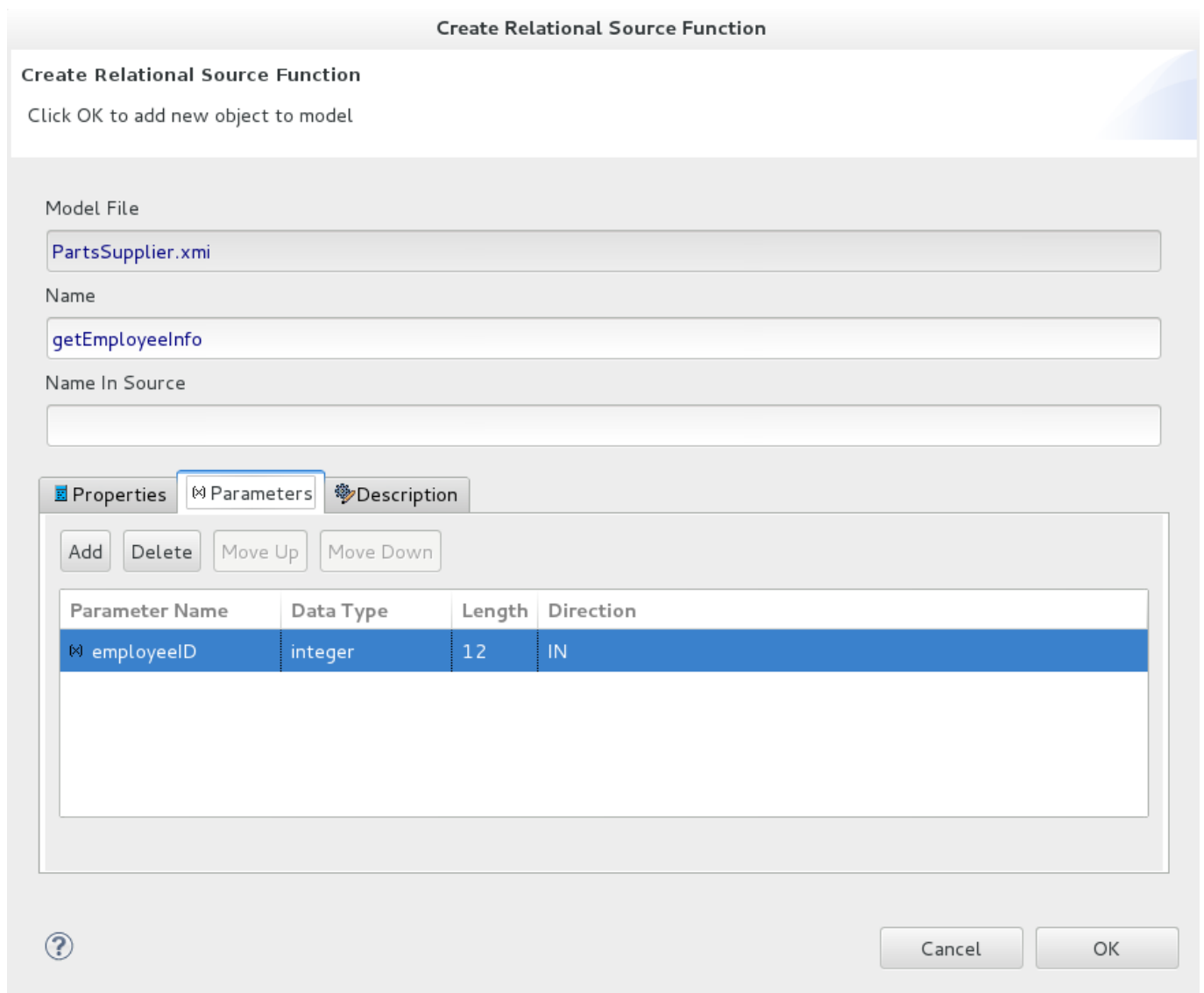


図12.3 Create New Source Function ダイアログ

終了したら、モデルのパッケージ図に新しいソース関数が表示されます。

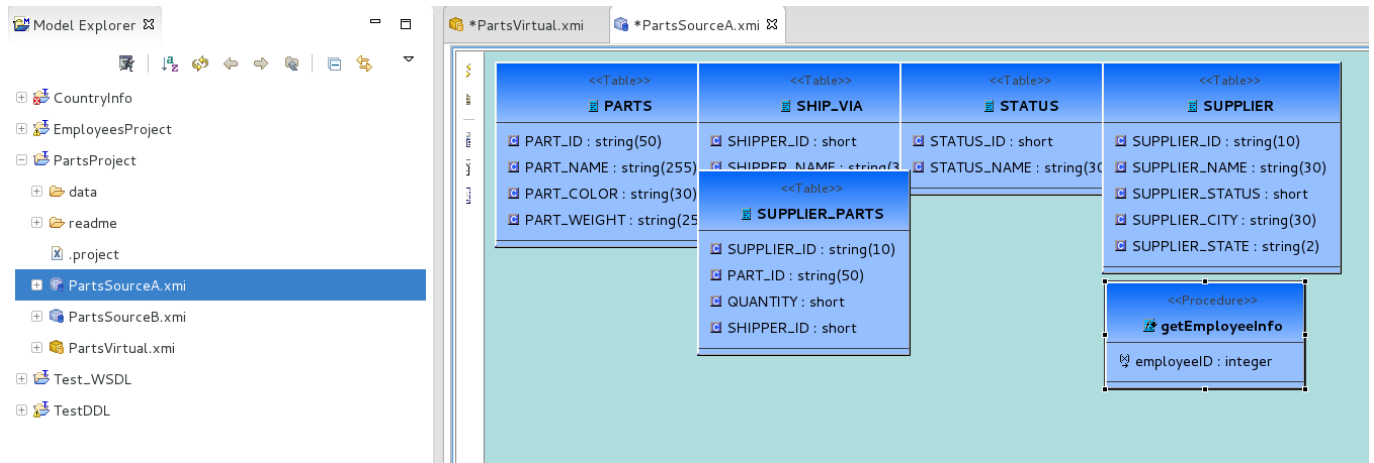


図12.4 パッケージ図の新しいソース関数

モデルの保存後、トランスフォーメーションで新しいソース関数を使用できるようになります。Expression Builder を開くと、モデルに対して名前が付けられたカテゴリーの Function ドロップダウンメニューでソース関数を選択できます。

### 12.1.2. リレーショナルテーブルウィザード

**New Child > Table...** と選択するとテーブルを作成できます。

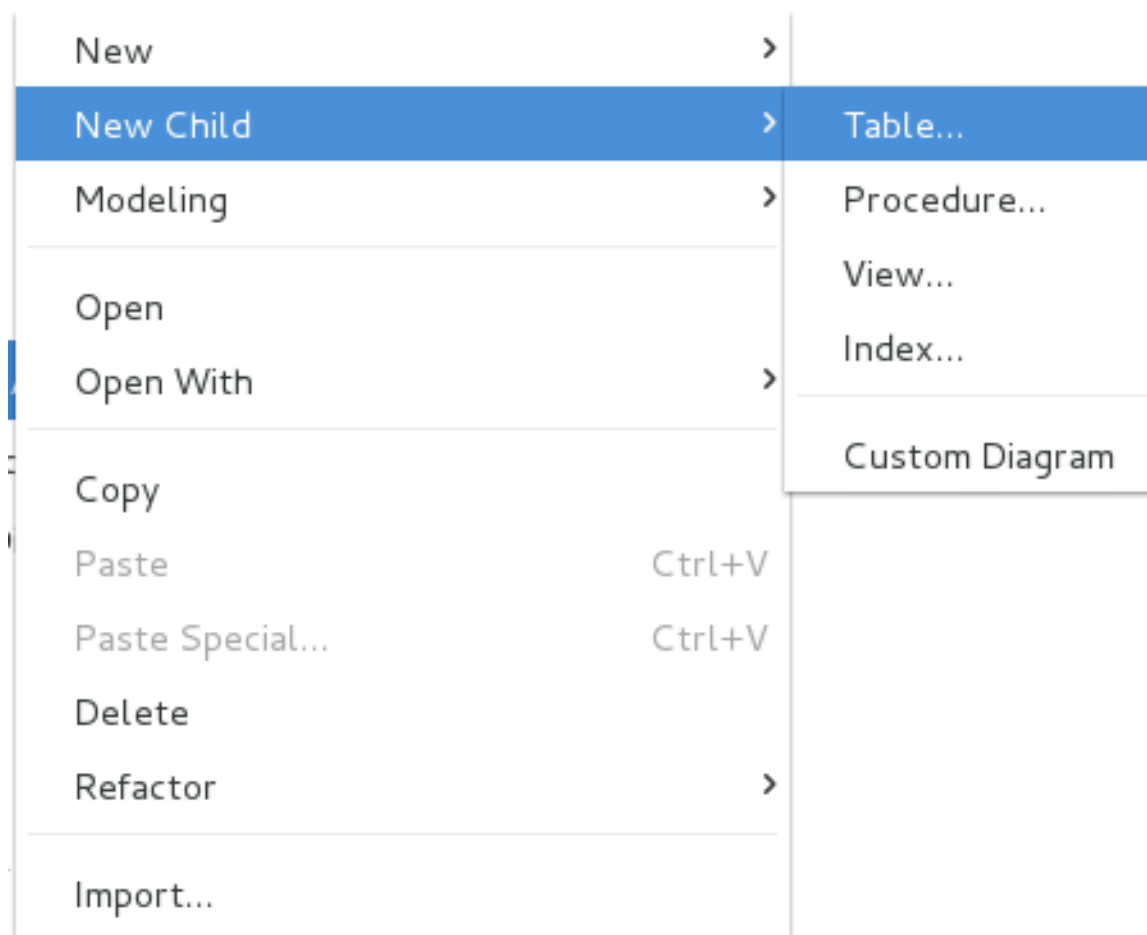


図12.5 新しいリレーショナルテーブルウィザード

このように選択すると、**Create Relational Table** ウィザードが表示されます。ウィザードのページには、リレーショナルテーブルの可能な定義を構成するさまざまなプロパティやコンポーネントを表す、タブ付きのパネルが含まれます。テーブル名を入力し、列、キー、制約、およびその他のオプションを定義した後に、**OK** をクリックします。

このウィザードは、リレーショナルテーブルの完全性や、テーブルおよびそのコンポーネントの検証状態を提供するのが目的です。ウィザードでは、編集集中にエラーが表示されることがありますが、エラーが含まれる不完全なテーブルを構築できるよう設計されています。

**Properties** というラベルが付いた最初のタブはシステムテーブルプロパティで、名前、ソースの名前、基数、「supports update」および「is system table properties」が含まれる簡単なテーブルプロパティの入力が含まれます。

The screenshot shows the 'Create Relational Table' wizard in Teiid Designer. The title bar reads 'Create Relational Table'. Below the title, there is a warning icon and the text 'No columns defined for table'. The main area contains several input fields: 'Model File' with 'PartsSupplier.xmi', 'Name' with 'PartsInfo', and 'Name In Source' with 'PartsInfo'. Below these is a tabbed interface with tabs for 'Properties', 'Columns', 'Primary Key', 'Unique Constraint', 'Foreign Keys', 'Indexes', and 'Native Query'. The 'Properties' tab is active, showing a 'Cardinality' field set to '0', and three checkboxes: 'Supports Update' (checked), 'Is System Table' (unchecked), and 'Materialized' (unchecked). There is a 'Materialized Table' field with an empty text box and a dropdown arrow. Below this is a large 'Description' text area. At the bottom right, there are 'Cancel' and 'OK' buttons. A help icon (?) is located at the bottom left.

図12.6 Properties タブ



**Columns** タブでは、基本的なリレーショナル列を作成および編集できます。これには、列の追加、削除、または移動が含まれ、名前、データタイプ、および長さプロパティの変更も含まれます。

**Create Relational Table**

**Create Relational Table**

Click OK to add new object to model

Model File  
PartsSupplier.xmi

Name  
PartsInfo

Name In Source  
PartsInfo

Properties
  **Columns**
 Primary Key
  Unique Constraint
  Foreign Keys
  Indexes
  Native Query

| Column Name                                  | Data Type | Length |
|--|-----------|--------|
| <input checked="" type="checkbox"/> ID       | string    | 10     |
| <input checked="" type="checkbox"/> name     | string    | 10     |
| <input checked="" type="checkbox"/> category | string    | 10     |
| <input checked="" type="checkbox"/> quantity | string    | 10     |
|  |           |        |

図12.7 Columns タブ

**Primary Key** タブでは、名前、ソースの名前、および列定義を編集できます。ボックスを消去するとデータが消去されることに注意してください。**Unique Constraint** タブには同一の情報が含まれます。

**Create Relational Table**

**Create Relational Table**  
Click OK to add new object to model

Model File  
PartsSupplier.xmi

Name  
PartsInfo

Name In Source  
PartsInfo

**Properties** | **Columns** | **Primary Key** | Unique Constraint | Foreign Keys | Indexes | Native Query

Include

Name

Name In Source

| Column Name                            |
|--|
| <input checked="" type="checkbox"/> ID |

図12.8 Primary Key タブ

**Foreign Keys** タブでは、複数の外部キーを作成、編集、および削除できます。

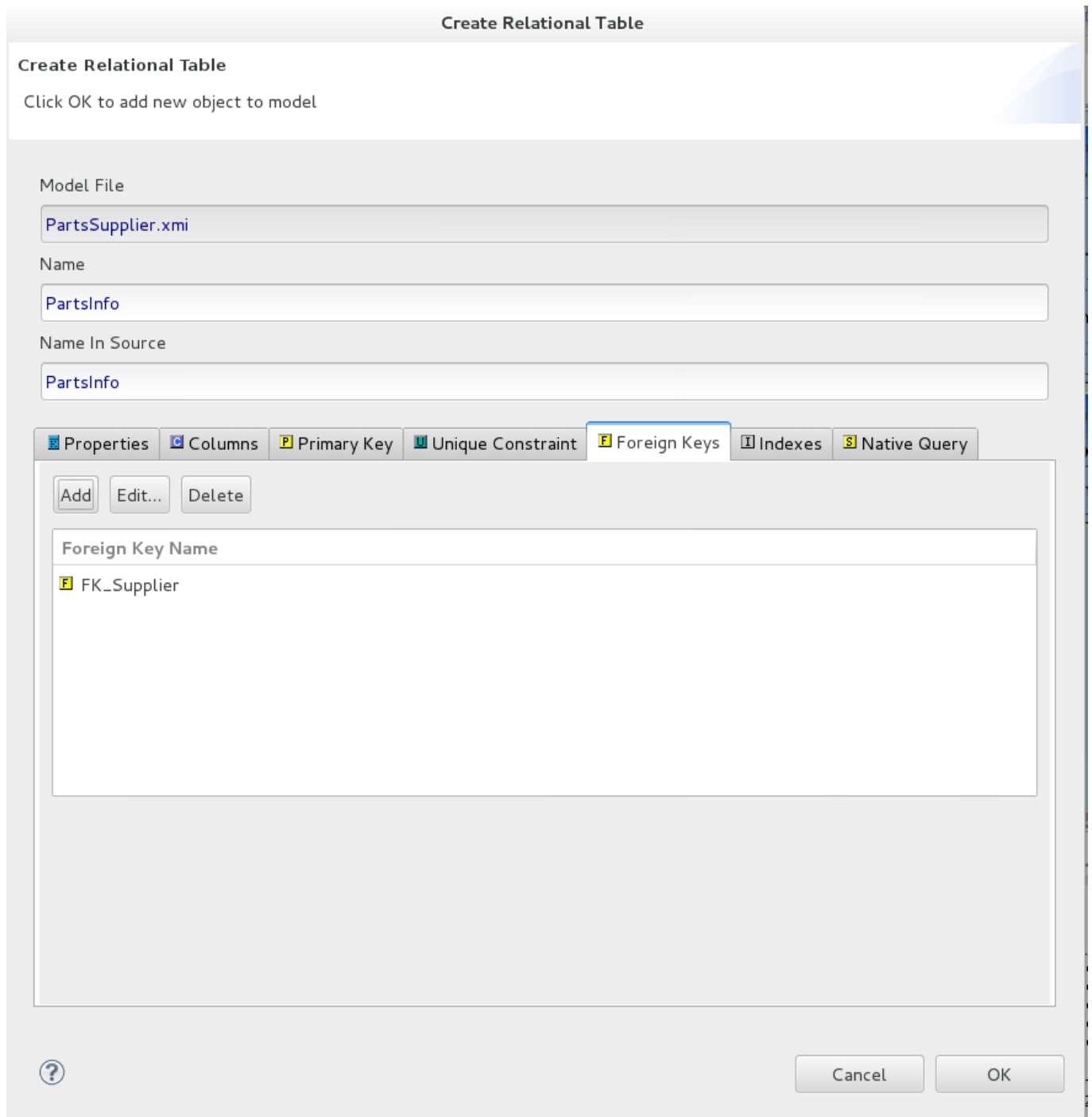


図12.9 Foreign Keys タブ

新しい **Foreign Key** を作成するには、**Add** ボタンを選択し、以下のテーブルでプロパティを入力または選択し、プロパティやキー参照を入力または選択します。Select Primary Key または Unique Constraint ターブルは選択されたリレーショナルモデルに存在する PK/UC を表示することに注意してください。モデルに PK または UC が含まれるテーブルがない場合、テーブルは空になります。

**Create Foreign Key**

**Create Foreign Key**  
Enter valid name, select a key or constraint, select column references and press OK

Name

Name In Source

Foreign Key Multiplicity

Unique Key Multiplicity

Allow Join

**Select Primary Key or Unique Constraint**

**P** PartsInfo: PK\_PartInfo

**Select Column References To FK**

**C** ID

**C** name

**C** category

図12.10 Create Foreign Key ダイアログ

## 12.2. リレーショナルビューのモデリング

### 12.2.1. マテリアライズドビューの作成

リレーショナルビューテーブルでは、以下のように Materialized プロパティを TRUE に設定し、Materialized Table 参照を設定するとマテリアライズドビューオプションを有効にできます。リレーショナルテーブルがすでに作成済みである必要があります。

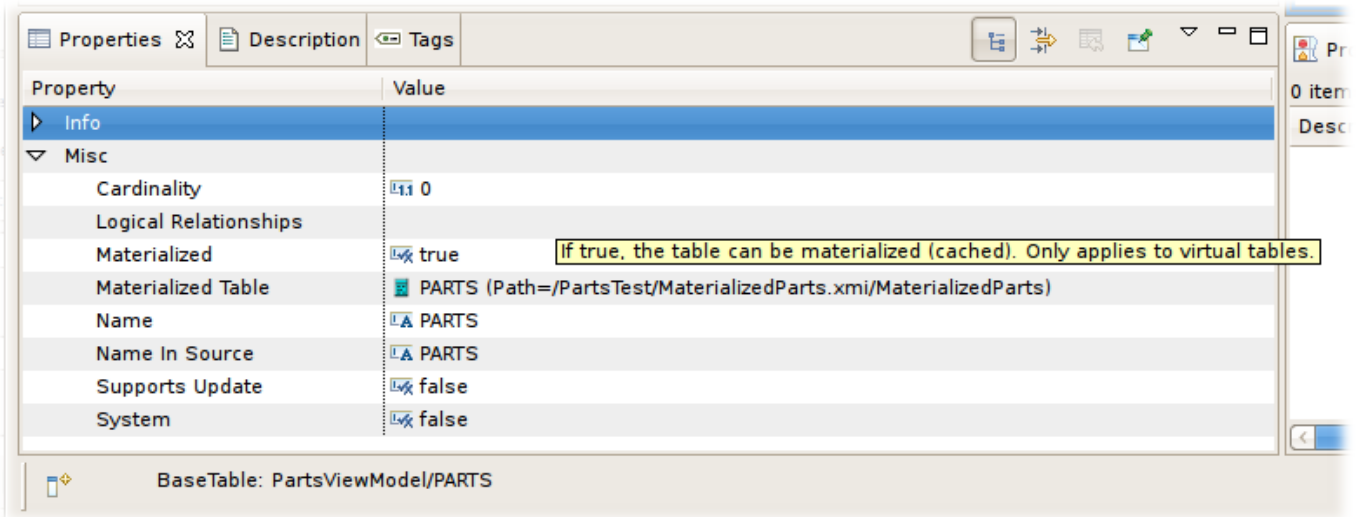


図12.11 Materialized テーブルプロパティ

**Teiid Designer** には、既存のビューテーブルを基にしたマテリアライズドリレーショナルテーブルを即座に作成できる機能が含まれています。

マテリアライズドビューを作成するには、以下を実行します。

1. マテリアライズプロセスは一度に1つのビューで実行されます。マテリアライズするビューを選択してから、**Model Explorer View**のビューテーブルを右クリックし、**Modeling > Create Materialized Views**と選択します。(2つ以上のテーブルを選択しないと、別のコンテキストメニューとウィザードが表示されることに留意してください。)

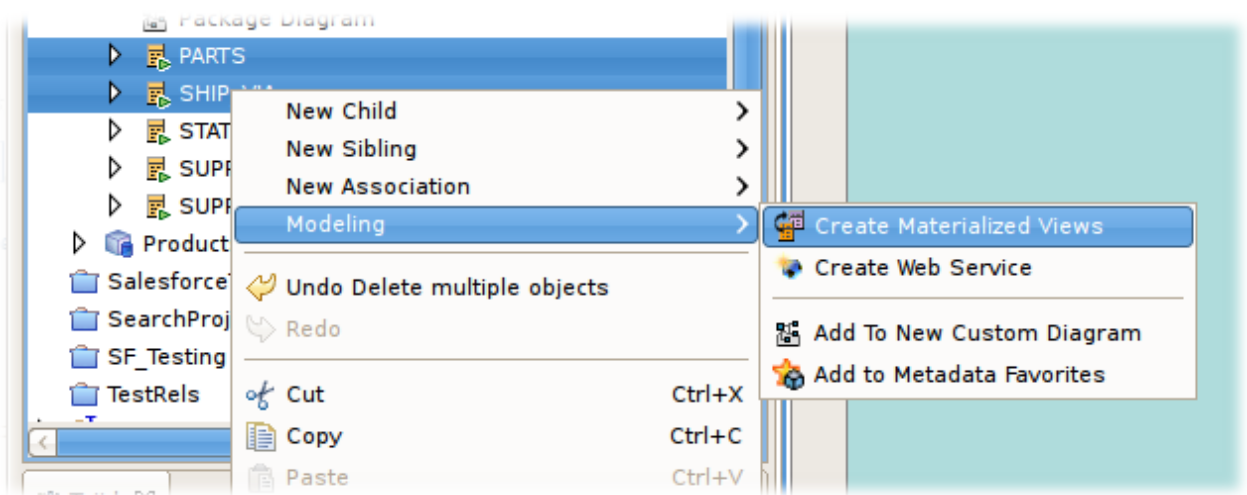


図12.12

2. **Create Materialized View Model** ダイアログで、生成されたリレーショナルテーブルの目的のリレーショナルモデルを指定または選択します。

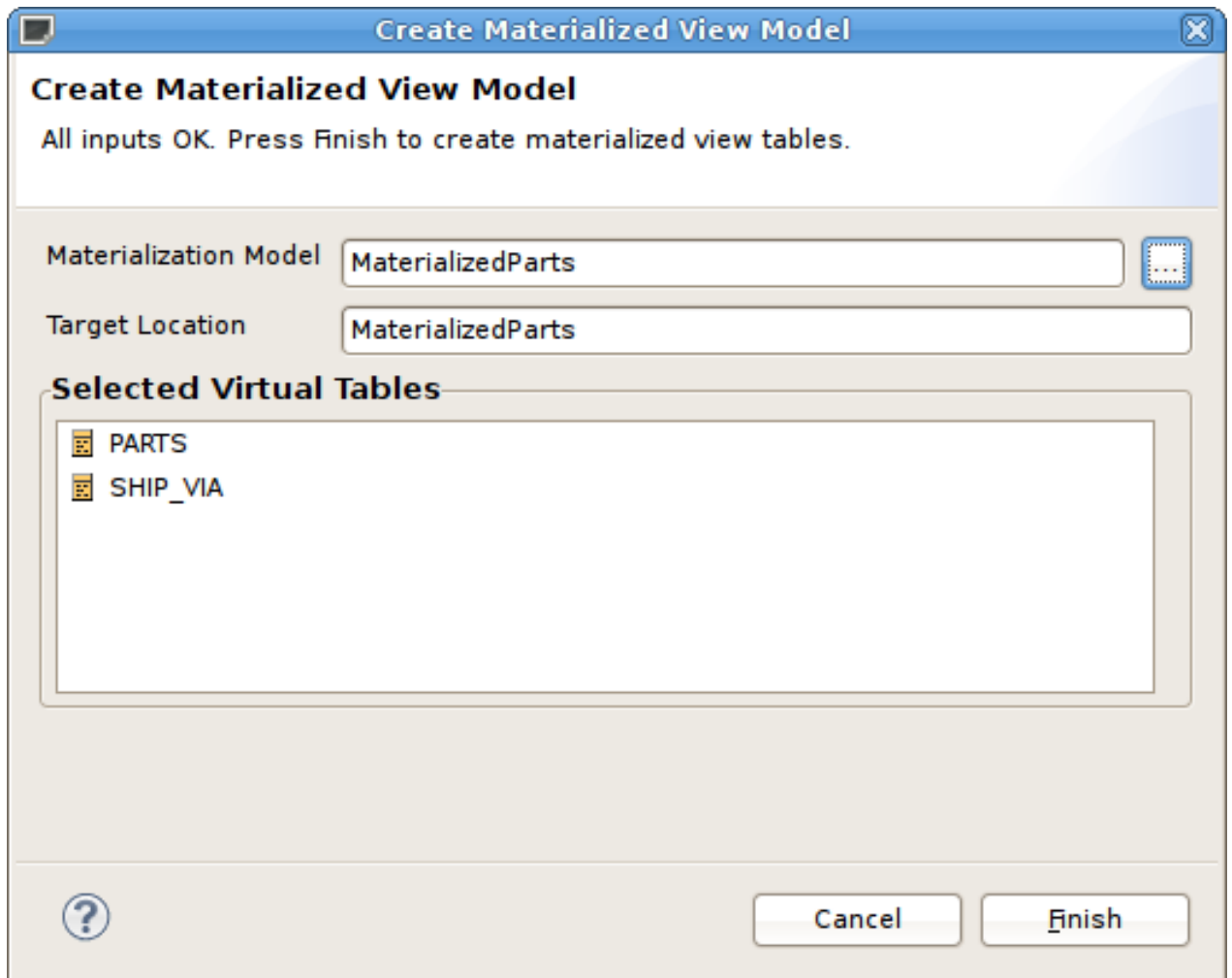


図12.13

3. **Browse ...** ボタンを選択すると **Relational Model Selector** ダイアログが表示されます。このダイアログでは、既存のリレーショナルモデルを選択したり、新しいモデルの一意名を指定したりできます。

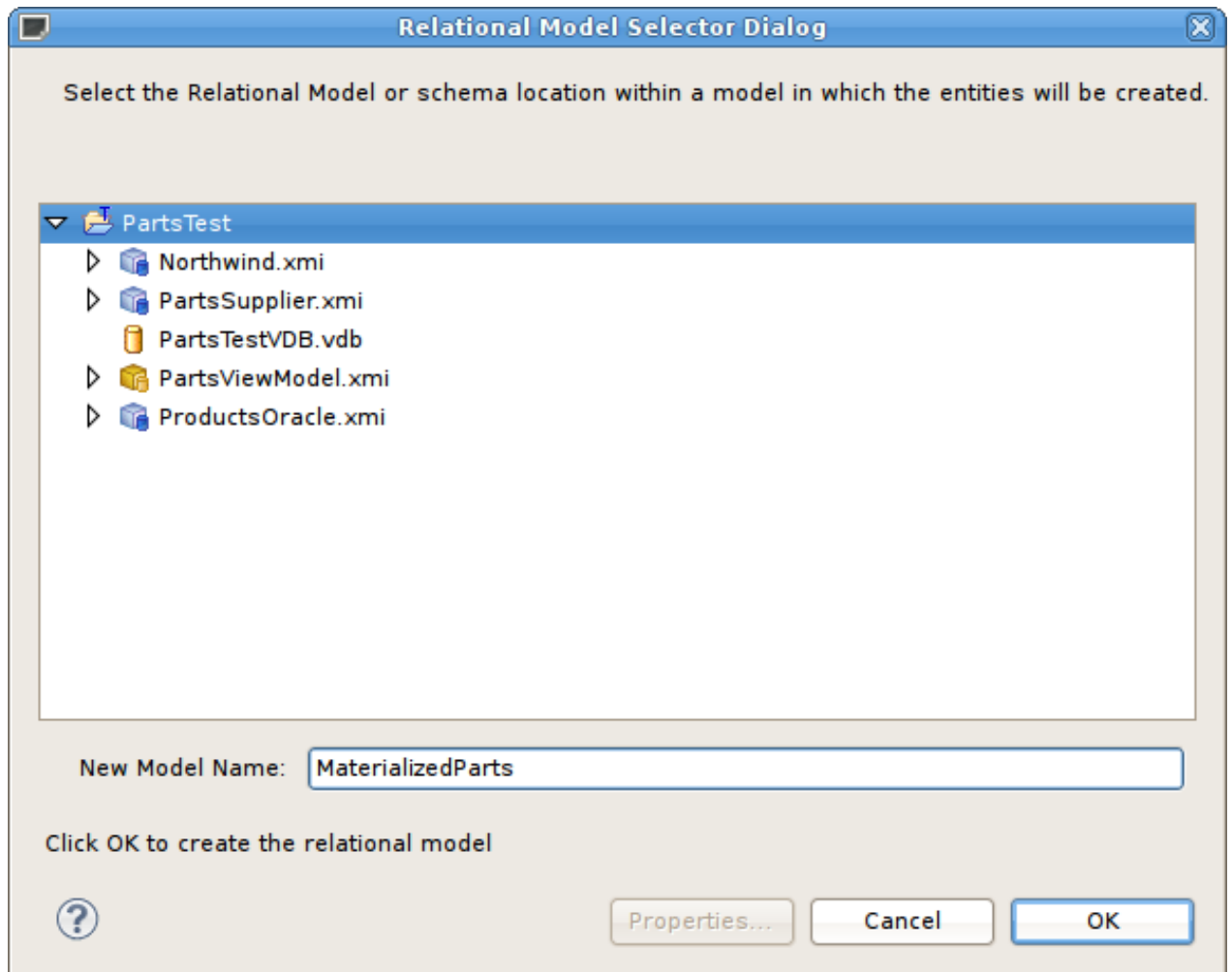


図12.14

4. **OK** をクリックして、選択したビューテーブルに対応するリレーショナルテーブルを作成します。  
Materialized プロパティを自動的に TRUE に設定し、Materialized Table 参照値を新たに生成されたテーブルに設定します。

終了後、ビューテーブルが新しい materialized プロパティで設定され、対応するリレーショナルテーブルがパッケージ図に表示されます。

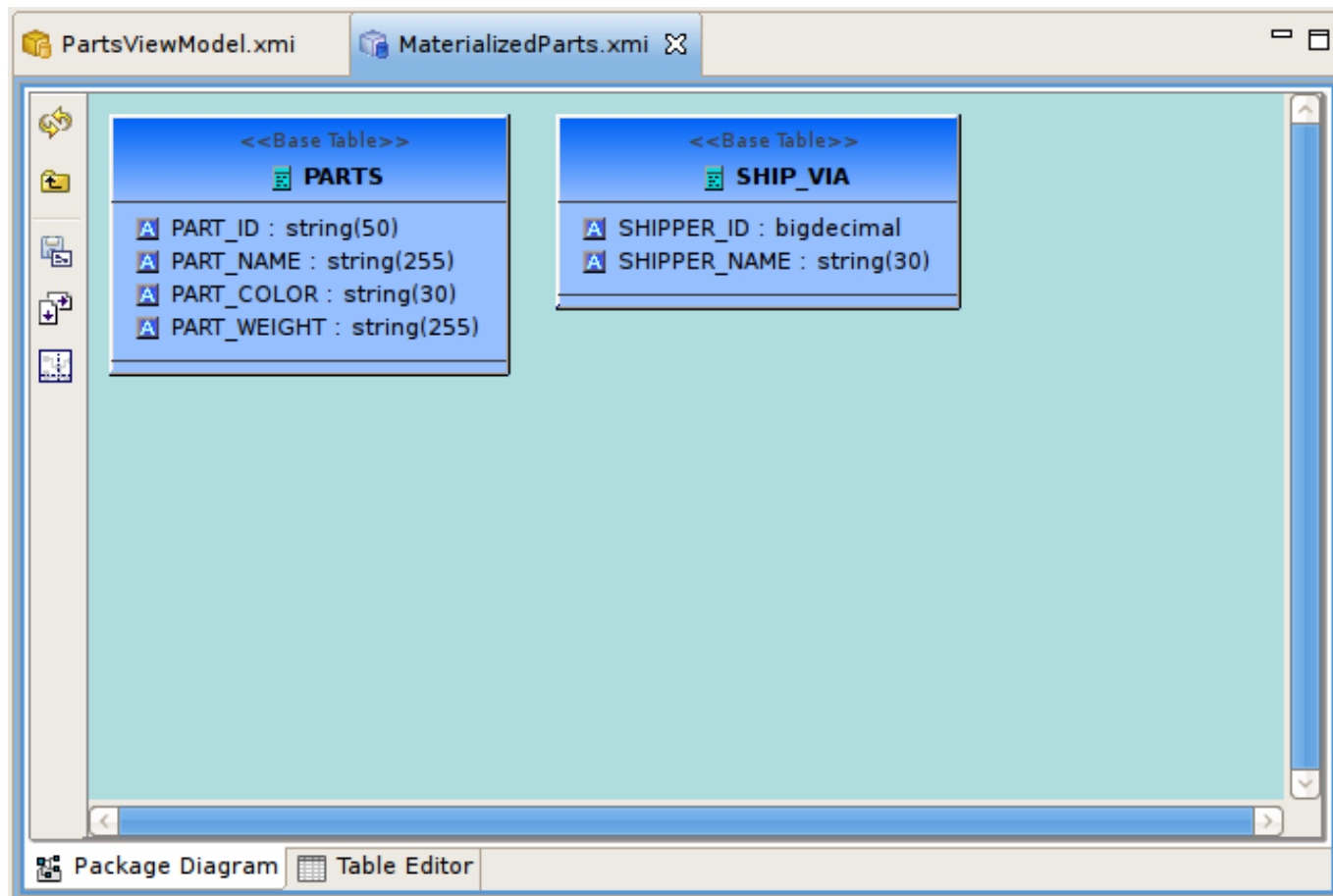


図12.15 Materialized テーブルプロパティ

### 12.2.2. JDG マテリアライズ

クライアント/サーバーモードで実行中の JDG キャッシュは、Red Hat JBoss Data Virtualization 固有の JCA コネクターをアクセス用のデータソースとして使用することができ、これはインストール中に WildFly 10.0.0 にデプロイされます。このコネクターは、Hot Rod クライアントを使用してリモートの JDG キャッシュにアクセスできるように設定可能です。

アクセスする先のキャッシュに "get(key)" を実行することで参照 pojo オブジェクトがある各 JDG キャッシュは、異なる設定のリソースアダプターを必要とします。

コネクターにおける JDG スキーマの設定には、protobuf 注釈またはマーシャラーのある protobuf (.proto) ファイルという 2つのオプションがあります。以下は要件になります。

- ※ 最小限 JDG 6.2 - この場合、pojo が JDG スキーマを設定するために、protobuf の定義ファイルと pojo マーシャラーを指定する必要があります。
- ※ 最小限 JDG 6.6 - これは、JDG スキーマの定義に使用する、定義済み protobuf 注釈が pojo にある場合に使用できます。

JDG キャッシュおよびアクセスする pojo へのマッピングを提供する以下のプロパティが必要になります。

表12.1 マッピング

| プロパティ名 | プロパティテンプレート | 説明 |
|--------|-------------|----|
|--------|-------------|----|



| プロパティ名       | プロパティテンプレート  | 説明   |
|--------------|--|--|
| CacheTypeMap | cacheName:className[;pkFieldName[:cacheKeyJavaType]] | 指定した cacheName で、ルート Java オブジェクト (pojo) クラス名をマップします。また、キャッシュに対するプライマリーキーとなるクラス属性を指定します。これはオプションではあるものの更新には必須となります。クラス属性タイプが異なる場合は、プライマリーキーの Java タイプを指定するオプションもあります。 |

JDG スキーマの設定に protobuf 定義ファイル (.proto) と pojo マーシャラーを使用する場合は、以下のプロパティが必要になります。

表12.2 ベース実行プロパティ

| プロパティ名                 | 必要性 | プロパティテンプレート   | 説明   |
|------------------------|-----|---|--|
| ProtobufDefinitionFile | はい  | jar にパッケージ化された Google Protobuf ファイルへのパス<br>(/quickstart/addressbook.proto など) | 該当なし   |
| MessageMarshaller      | はい  | marshaller \[,marshaller,..]  | 各メッセージマーシャラーがマップされたクラス名を含みません (class:marshaller,\[class:marshaller,..])。シリアル化のために登録されます。 |
| MessageDescriptor      | はい  | 該当なし  | キャッシュ内のルートオブジェクトに対するメッセージ記述子クラス名   |

pojo クラスは、キャッシュにデータ保存する際に使用するオブジェクトです。

スキーマの定義に pojo を使用する場合は、protobuf 注釈を使用してください。

protobuf 定義とマーシャラーを使用する場合は、これらを jar にパッケージ化してください (または、classpath に含まれる別個の jar)。

クラスは jar にパッケージ化して、モジュールとしてデプロイできるようにします。

```
public class Person {

    @ProtoField(number = 2, required = true)
    public String name;
    @ProtoField(number = 1, required = true)
    public int id;
    @ProtoField(number = 3)
    public String email;
    private List<PhoneNumber> phones;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

```

    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public List<PhoneNumber> getPhones() {
        return phones;
    }

    public void setPhones(List<PhoneNumber> phones) {
        this.phones = phones;
    }
}

```

リソースアダプターが pojo を使用するように設定するには、pojo jar をモジュールとして JBOSS EAP サーバーにデプロイします。

RemoteCacheManager の作成およびアクセス方法を定義するには、以下のいずれかのプロパティーが必要になります。

**表12.3 設定**

| プロパティー名      | 必要性 | プロパティーテンプレート   | 説明   |
|--------------|-----|--|--|
| CacheTypeMap | はい  | cacheName:className[:pkFieldName[:cacheKeyJavaType]] | ルート Java オブジェクトクラス名をキャッシュにマップします。また、キャッシュに対するプライマリーキーとなるクラス属性を指定します。これはオプションではあるものの更新には必須となります。クラス属性タイプが異なる場合は、プライマリーキーの Java タイプを指定します。 |
| ProtobinFile | はい  | 該当なし   | jar にパッケージ化された Google Protobin ファイルへのパス<br>(/quickstart/addressbook.protobin など)   |

| プロパティ名                     | 必要性 | プロパティテンプレート                 | 説明  |
|----------------------------|-----|-----------------------------|---|
| MessageMarshallers         | はい  | marshaller [,marshaller,..] | 各メッセージマーシャラーがマップされたクラス名を含みます (class:marshaller, [class:marshaller,..])。シリアル化のために登録されます。 |
| MessageDescriptor          | はい  | 該当なし                        | キャッシュ内のルートオブジェクトに対するメッセージ記述子クラス名  |
| モジュール                      | なし  | 該当なし                        | ロードする必要があるキャッシュクラスを含む JBoss EAP モジュールを指定します。  |
| CacheJndiName              | なし  | 該当なし                        | CacheContainer を見つけるための JNDI 名  |
| RemoteServerList           | なし  | host:port[;host:port....]   | キャッシュにアクセスするために一緒にクラスタ化するホストおよびポートを指定します。   |
| HotRodClientPropertiesFile | なし  | 該当なし                        | リモートキャッシュに対する接続を設定するための HotRod プロパティファイル。   |

protobuf 定義ファイルおよびマーシャラーを使用する場合は、以下のプロパティを定義します。

表12.4 プロパティ

| プロパティ名 | 必要性 | プロパティテンプレート | 説明   |
|--------|-----|-------------|--|
| モジュール  | なし  | 該当なし        | ロードする必要があるキャッシュクラスを含む WildFly モジュールを指定します。 |

外部マテリアライズにリモートキャッシュを使用する場合は、以下のプロパティの設定が必要になります。

表12.5 マッピング

| プロパティ名           | 必要性 | 説明   |
|------------------|-----|--|
| StagingCacheName | はい  | マテリアライズに使用するステージングキャッシュのキャッシュ名   |
| AliasCacheName   | はい  | マテリアライズに使用するキャッシュのエイリアス追跡に使用するエイリアスキャッシュのキャッシュ名。このキャッシュは、他の設定済みマテリアライズと共有できます。 |

データソースを作成するには多くの方法があります (CLI、AdminShell、管理コンソールなどを使用)。最初の例は、Protobuf 定義およびマーシャラーを定義する JDG remote-query quick start:me Sample Resource Adapter への接続に使用されるリソースアダプターの XML スニペットです。

```
<resource-adapter id="infinispanRemQS">
  <module slot="main" id="org.jboss.teiid.resource-adapter.infinispan.hotrod"/>
  <connection-definitions>
    <connection-definition class-
```

```

name="org.teiid.resource.adapter.infinispan.hotrod.InfinispanManagedConnectionFactory" jndi-name="java:/infinispanRemote" enabled="true" use-java-context="true" pool-name="infinispanDS">
    <config-property name="CacheTypeMap">

addressbook:org.jboss.as.quickstarts.datagrid.hotrod.query.domain.Person;id
    </config-property>
    <config-property name="ProtobufDefinitionFile">
        /quickstart/addressbook.proto
    </config-property>
    <config-property name="MessageDescriptor">
        quickstart.Person
    </config-property>
    <config-property name="Module">
        com.client.quickstart.pojos
    </config-property>
    <config-property name="MessageMarshallers">
org.jboss.as.quickstarts.datagrid.hotrod.query.domain.Person:org.jboss.as.quickstarts.datagrid.hotrod.query.marshallers.PersonMarshaller,org.jboss.as.quickstarts.datagrid.hotrod.query.domain.PhoneNumber:org.jboss.as.quickstarts.datagrid.hotrod.query.marshallers.PhoneNumberMarshaller,org.jboss.as.quickstarts.datagrid.hotrod.query.domain.PhoneType:org.jboss.as.quickstarts.datagrid.hotrod.query.marshallers.PhoneTypeMarshaller
    </config-property>
    <config-property name="RemoteServerList">
        127.0.0.1:11322
    </config-property>
    </connection-definition>
</connection-definitions>
</resource-adapter>

```

以下は、注釈付き Pojo を使用したリソースアダプターのサンプルです。

```

<resource-adapter id="infinispanRemQSDSL">
    <module slot="main" id="org.jboss.teiid.resource-adapter.infinispan.dsl"/>
    <connection-definitions>
        <connection-definition class-
name="org.teiid.resource.adapter.infinispan.dsl.InfinispanManagedConnectionFactory" jndi-name="java:/infinispanRemoteDSL" enabled="true" use-java-context="true" pool-name="infinispanRemoteDSL">
            <config-property name="RemoteServerList">
                127.0.0.1:11322
            </config-property>
            <config-property name="CacheTypeMap">

addressbook_indexed:org.jboss.as.quickstarts.datagrid.hotrod.query.domain.Person;id
                </config-property>
            </connection-definition>
        </connection-definitions>
    </resource-adapter>

```

以下は、外部マテリアライズ向けのリソースアダプターのサンプルです。

```

        <resource-adapter id="infinispanRemQSDSL">
            <module slot="main" id="org.jboss.teiid.resource-
adapter.infinispan.hotrod"/>
            <connection-definitions>
                <connection-definition class-
name="org.teiid.resource.adapter.infinispan.hotrod.InfinispanManagedConnecti
onFactory" jndi-name="java:/infinispanRemoteDSL" enabled="true" use-java-
context="true" pool-name="infinispanRemoteDSL">
                    <config-property name="CacheTypeMap">

addressbook_indexed:org.jboss.as.quickstarts.datagrid.hotrod.query.domain.Pe
rson;id

                    </config-property>
                    <config-property name="StagingCacheName">
                        addressbook_indexed_mat
                    </config-property>
                    <config-property name="AliasCacheName">
                        aliasCache
                    </config-property>
                    <config-property name="Module">
                        com.client.quickstart.addressbook.pojos
                    </config-property>
                    <config-property name="RemoteServerList">
                        127.0.0.1:11322
                    </config-property>
                </connection-definition>
            </connection-definitions>
        </resource-adapter>

```

HotRod トランスレーターはタイプ ispn-hotrod で知られており、java オブジェクトをリモートの JDG Cache からシリアル化のために Google Protobuf を使用して Hot Rod クライアント経由で読み取ることができます。これにより Teiid が JDG DSL を使用してリモートキャッシュにクエリできるようになります。このトランスレーターは Object Translator を拡張し、オブジェクトの読み取りおよび書き込みのコアプロセスにこれを使用します。また、以下のことが実行可能です。

- ✧ キャッシュからオブジェクトを取得し、行と列に変換します。
- ✧ キャッシュへの書き込みをサポートします。
- ✧ 外部マテリアライズを使用してクエリパフォーマンスを改善します。

コネクタには以下のエイパビリティーがあります。

- ✧ Compare Criteria - EQ
- ✧ Compare Criteria Ordered - LT, GT, LE, GE - SupportsComapareCriteriaOrdered のサポートはアクセスしている JDG のバージョンで制御されます。JDG バージョン 6.5 およびそれ以前は、JDG の問題のためにこれが false に設定されます。
- ✧ And/Or Criteria
- ✧ In Criteria
- ✧ Like Criteria
- ✧ Order By

- ※ INSERT, UPDATE, DELETE (トランザクション以外)

以下のものは処理向けに JDG ヘプッシュされませんが、Red Hat JBoss Data Virtualization 内で実行されません。

- ※ Not (NE)
- ※ IsNull

以下の制限があります。

- ※ 'Not' のサポートは無効になっています。
- ※ boolean data type: 挿入の値が指定されていない場合、または protobuf 定義ファイルでデフォルト値が定義されていない場合は JDG が例外をスローします。
- ※ char data type: Protobuf データタイプでサポートされているタイプではありません (<https://developers.google.com/protocol-buffers/docs/proto#scalar>)。protobuf マーシャラーで変換を処理するか、char のデータタイプで Teiid ビューを作成する必要があります。
- ※ 1-to-Many が現在サポートしているのは Collection または Array のみで、Maps はサポートされていません。
- ※ 書き込みトランザクションは、Hot Rod クライアント使用時には JDG でサポートされません。

キャッシュ内のオブジェクトを表すメタデータの定義には、いくつかのオプションがあります。Red Hat では、Teiid Designer 内の Teiid Connection Importer を使用して、オブジェクトキャッシュに基づく物理ソースモデルを作成することを推奨しています。テーブル列は、登録済みクラスに対応する google protobuf 定義から作成されます。

Teiid Designer を使用してオブジェクトキャッシュに基づく物理ソースモデルを手動で作成するには、以下の定義要件を使用します。

```
<model name="People" type="Physical">
  <property name="importer.useFullSchemaName" value="false"/>

  <source name="infinispan-hotrod-connector" translator-name="ispn-hotrod" connection-jndi-name="java:/infinispanRemoteDSL" />
</model>
```

メタデータは、キャンセル内の定義済みオブジェクトのリバースエンジニアリングで解決されます。これは物理ソースモデル構築で Teiid Designer Teiid Connection Importer を使用している場合に便利です。

DDL を使用してメタデータを定義することもできます。Object Translator の例を参照してください。

ベースの定義要件については、Object Translator Metadata セクションを参照してください。

列の protobuf 定義がインデックス化されている、または pojo クラスが属性/メソッドの注釈付きの場合は、列は SEARCHABLE として特定されます。

1-to-\* 関係クラスには root クラス/表にマッピングする外部キーが必要になります。ここでは、外部キーのソース内の名前は、それらの子オブジェクトにアクセスするための root クラスメソッド名になります。これはクラスメソッドであって、Google protobuf 定義の参照ではないことに注意してください。

コンテナ/子クラスは、NIS にピリオドが含まれる属性を持つことになります。これは、これが Google protobuf 定義と DSL クエリで使用されることが予想されるものにマッピングするためです。

このトランスレーターは、外部実体化のためのキャッシュ使用をサポートします。ただし、[Infinispan-HotRod resource-adapter] とトランスレーターで特定の設定変更が必要になります。

外部実体化は、BEFORE\_LOAD\_SCRIPT と AFTER\_LOAD\_SCRIPT でのネイティブクエリの使用で有効になります。トランスレータの上書きは、ネイティブクエリ SupportsNativeQueries=true を有効に設定する必要があります。

以下のマテリアライズプロパティを定義する必要があります。

表12.6 マッピング

| スクリプト                                | ネイティブクエリ         | 説明   |
|--------------------------------------|------------------|--|
| teiid_rel:MATVIEW_BEFORE_LOAD_SCRIPT | truncate cache   | staging cache として特定されたキャッシュを切り詰めます。                      |
| teiid_rel:MATVIEW_AFTER_LOAD_SCRIPT  | swap cache names | キャッシュのエイリアスをスワップし、プラーキーキャッシュが最近読み込まれたキャッシュをポイントするようにします。 |

以下の例は、DDL の読み込みスクリプトを定義します。

```
..
"teiid_rel:MATVIEW_BEFORE_LOAD_SCRIPT" 'execute
StockMatCache.native('truncate cache');',
"teiid_rel:MATVIEW_LOAD_SCRIPT" 'insert into StockMatCache.Stock (productId,
symbol, price, companyName) SELECT A.ID, S.symbol, S.price, A.COMPANY_NAME
FROM Stocks.StockPrices AS S, Accounts.PRODUCT AS A WHERE S.symbol =
A.SYMBOL',
"teiid_rel:MATVIEW_AFTER_LOAD_SCRIPT" 'execute StockMatCache.native('swap
cache names');',
```

RDBMS を使用して表の名前を変更する方法をシミュレートするためにネイティブクエリが使用されます。これは、JDG が現在キャッシュの名前変更をサポートしていないためです。このため、ネイティブクエリが "staging" キャッシュのクリーンアップと、キャッシュエイリアスのスワップを開始します。

また、ネイティブクエリの実行は直接クエリ手順のサポートを通じて行われます。実行される手順は、ネイティブと呼ばれます。



### 警告

この機能は、ソースに対してコマンドを実行するというセキュリティリスクがあるため、デフォルトではオフになっています。この機能を有効にするには、SupportsDirectQueryProcedure と呼ばれる実行プロパティ [Override Execution Properties] を true に上書きします。

Teiid Designer 内のキャッシュテーブルを手動でモデル化する場合は、プロパティ "primary\_table" の定義のためにプロパティ拡張子を追加する必要があります。

```
SET NAMESPACE 'http://www.teiid.org/translator/object/2016' AS n0;

CREATE FOREIGN TABLE Trade (
    ....
    CONSTRAINT PK_TRADEID PRIMARY KEY(tradeId)
) OPTIONS (UPDATABLE TRUE);

CREATE FOREIGN TABLE ST_Trade (
    ....
```

```
) OPTIONS (NAMEINSOURCE 'Trade', UPDATABLE TRUE, "n0:primary_table"
'ObjectSchema.Trade');
```



**注記**

手動でのモデリングを行う場合は、列名が POJO のものと一致する必要があります。

このトランスレーターの JDG HotRod Data Sources リソースアダプターを参照してください。JNDI、server list、または Hot Rod プロパティーでキャッシュコンテナをルックアップするように設定できます。



**重要**

テーブルおよび列名の作成には、(Teiid Connection Importer や Materialize といった) Teiid Designer ツーリングウィザードの使用が強く推奨されます。これではなく、JDG リソースアダプターをポイントする VDB XML フォームを使用する場合は、DDL メタデータを定義するのではなく、トランスレーターがメタデータを公開するようにしてください。

これを行う理由は、JDG スキーマが大文字と小文字を区別するためです。JDG へのクエリに使用される 'name' は、使用されたプロパティーの名前 (methodName) に一致する必要があります。

1. 注釈の使用において、pojo メソッド名はクラスを設定ビルダーに渡すことで、JDG スキーマの設定に使用されます。
2. protobuf/mashallers の使用において、は、JDG スキーマは protobuf ファイルを使用して設定されます。

この両方の設定方法で、JDG キャッシュが記述子を提供するようになります。列名 (および複雑なオブジェクトの使用時には nameInSources ) はこれより派生します。モデルで使用する必要があるのは、これらの名前です。

**12.2.3. リレーショナルビューテーブルウィザードの作成**

**New Child > Table...** とクリックすると、リレーショナルビューテーブルを作成できます。またこのアクションには、目的の列および SQL トランスフォーメーションでのビューテーブルの作成が含まれます。

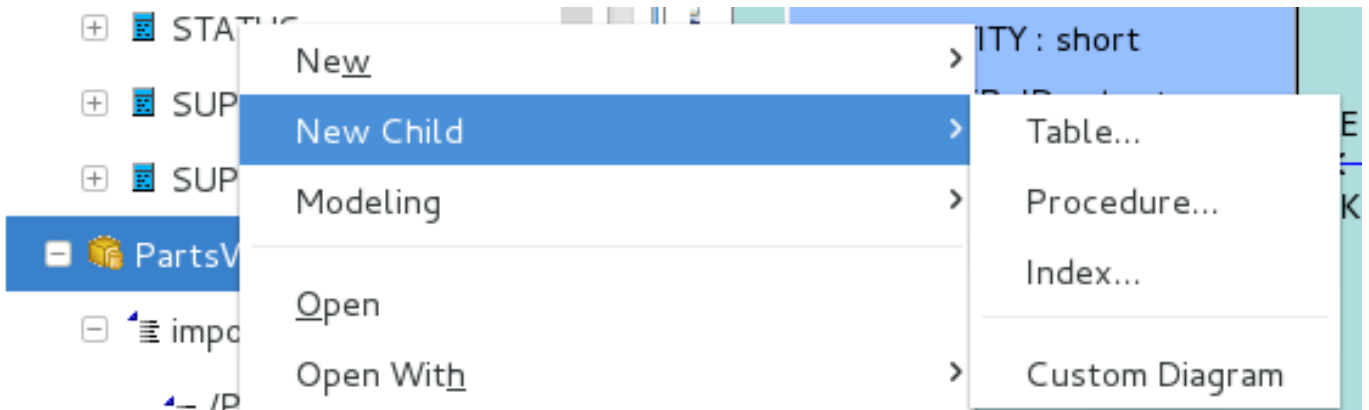


図12.16 新しいビューテーブルウィザード



このように選択すると、**Create Relational View Table** ウィザードが表示されます。ウィザードのページには、リレーショナルビューテーブルの可能な定義を構成するさまざまなプロパティやコンポーネントを表す、タブ付きのパネルが含まれます。テーブル名の入力、列の定義、および SQL トランスフォーメーションの指定を行った後、**OK** をクリックします。

このウィザードは、リレーショナルビューテーブルの完全性や、テーブルおよびそのコンポーネントの検証状態を提供するのが目的です。ウィザードでは、編集集中にエラーや警告が表示されることがありますが、テーブルの定義が不完全でも終了できるよう設計されています。

**Properties** というラベルが付いた最初のタブはシステムテーブルプロパティで、名前および「supports update」が含まれる簡単なテーブルプロパティの入力が含まれます。

The screenshot shows the 'Create Relational View Table' wizard with the 'Properties' tab selected. The title bar reads 'Create Relational View Table'. Below the title, there is a warning icon and the text 'No columns defined for table'. The 'Model File' field contains 'PartsVirtual.xmi'. The 'Name' field contains 'myTable'. The 'Name In Source' field is empty. The 'Properties' tab is active, showing a 'Cardinality' field set to '0', a checked 'Supports Update' checkbox, an unchecked 'Is System Table' checkbox, and an unchecked 'Materialized' checkbox. The 'Table Reference' field is empty with a dropdown arrow. Below is a large 'Description' text area. At the bottom, there are 'Cancel' and 'OK' buttons.

図12.17 Properties タブ

**Columns** タブでは、基本的なリレーショナル列を作成および編集できます。これには、列の追加、削除、または移動が含まれ、名前、データタイプ、および長さプロパティの変更も含まれます。

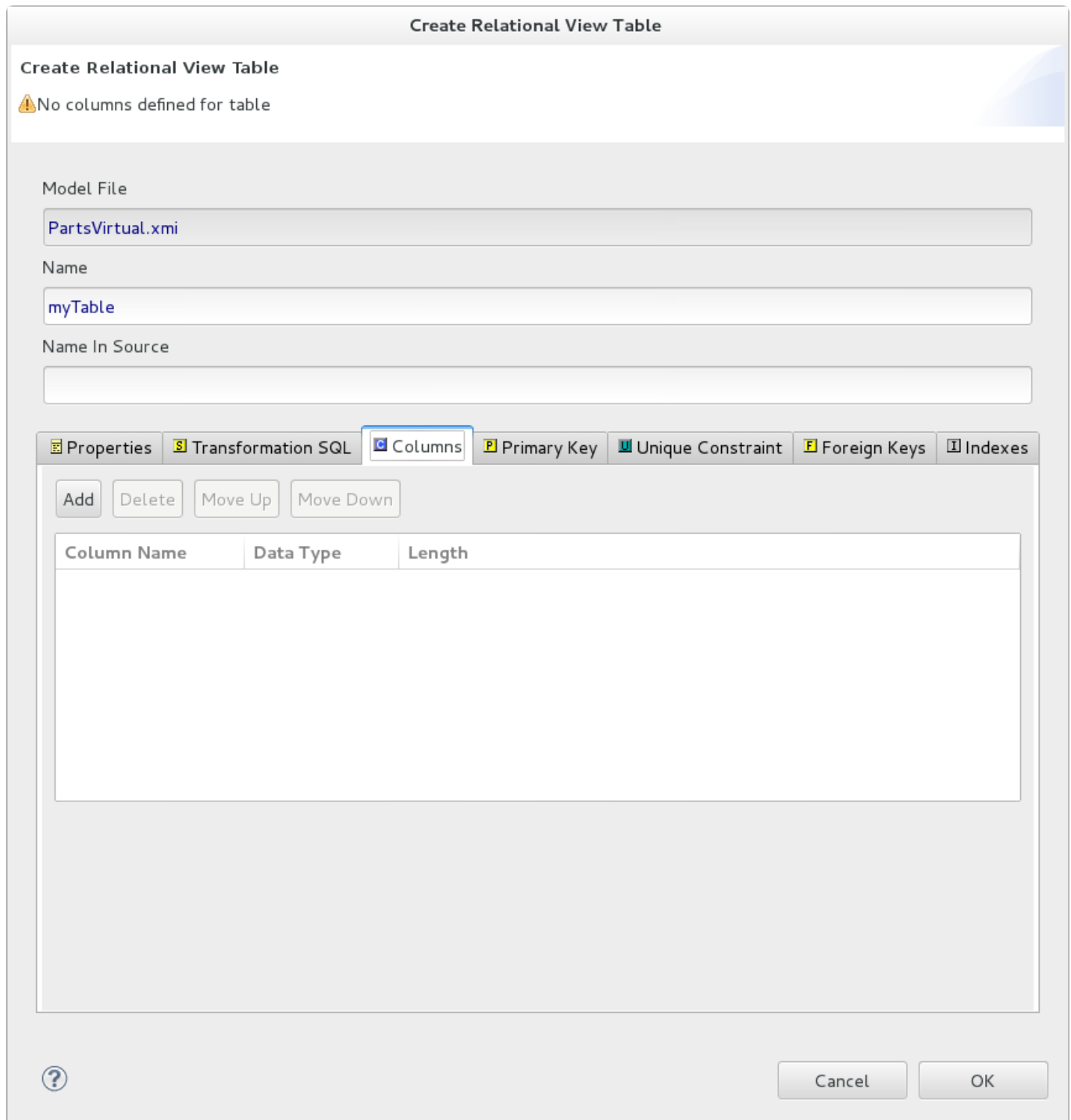


図12.18 Columns タブ

**Transformation SQL** タブでは、リレーショナルビューの SQL トランスフォーメーションの編集ができます。希望の SQL を直接テキストエリアに入力することができます。または、**Select SQL Template** ボタンをクリックして SQL テンプレートを選択することもできます。

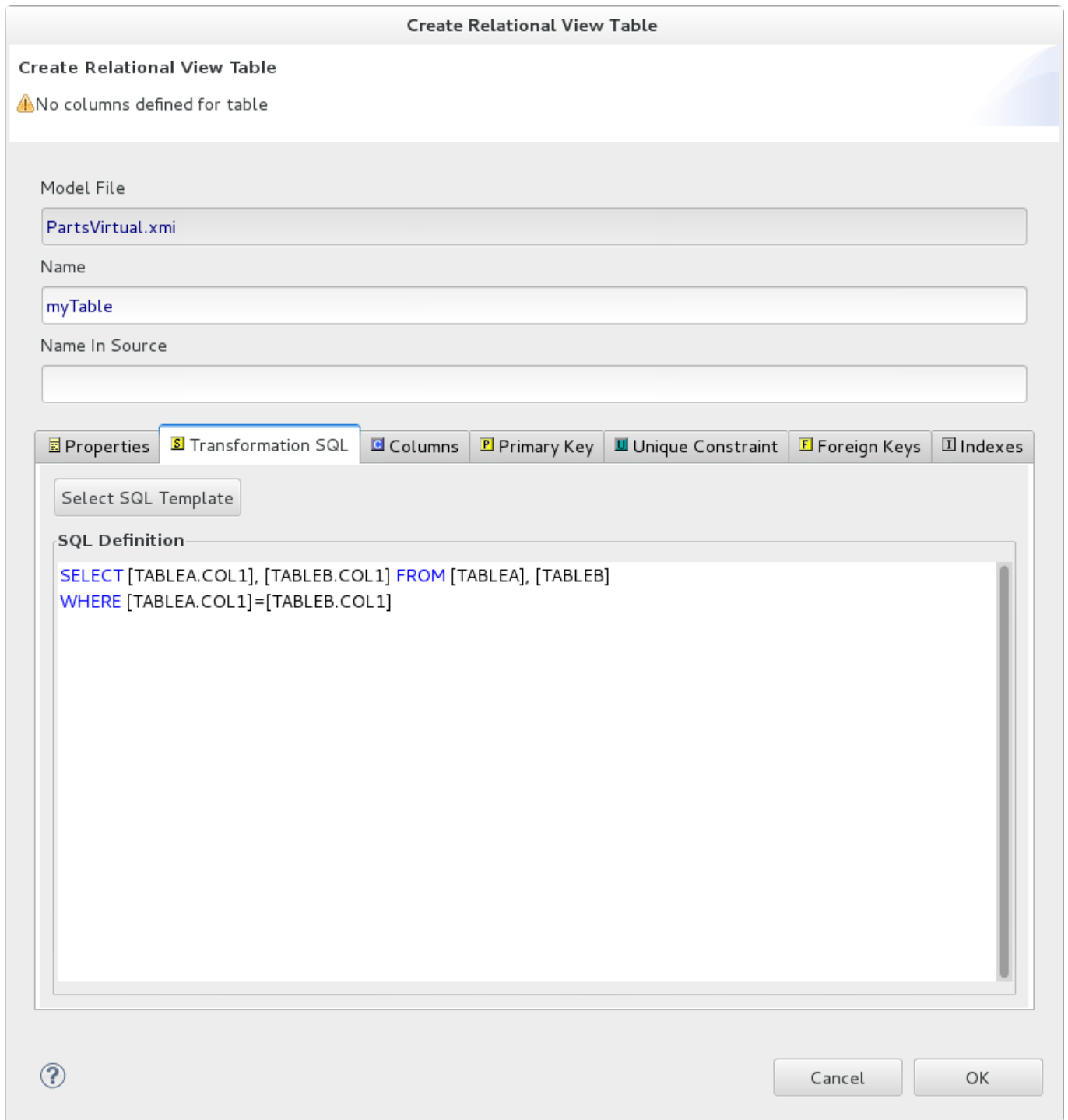


図12.19 Transformation SQL タブ

Transformation SQL タブで **Select SQL Template** ボタンが選択されると、**Choose a SQL Template** ダイアログが表示されます。

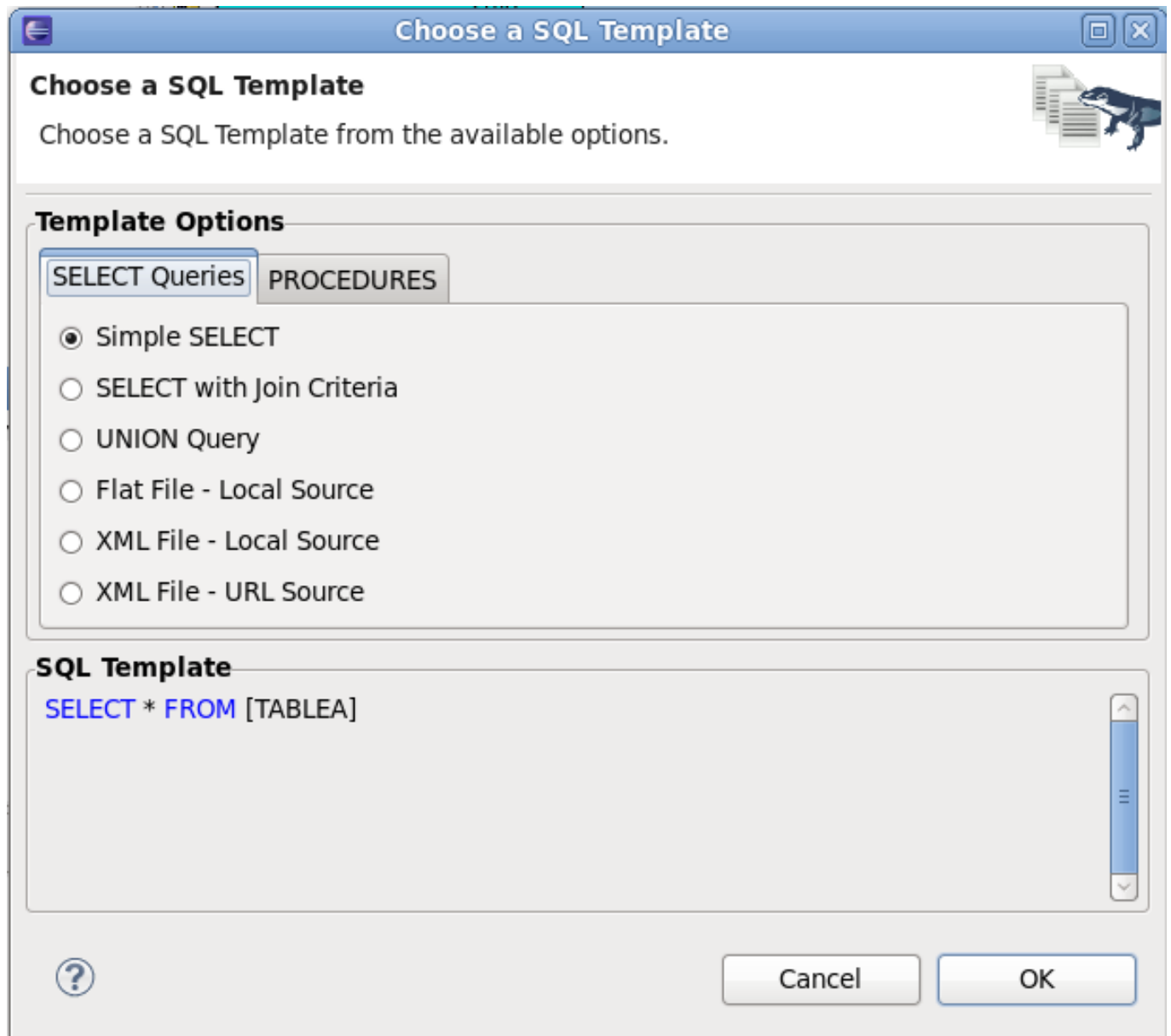


図12.20 SQL Templates ダイアログ

ダイアログを使用して一般的な SQL テンプレートを選択できる可能性があります。使用コンテキストによっては、**SELECT** クエリータブの他に **PROCEDURES** タブも使用できることがあります。**Transformation Editor** にて、右クリックで表示されるコンテキストメニューから **SQL Template** ダイアログにアクセスすることもできます。

## 12.3. XML ドキュメントのモデリング

### 12.3.1. スキーマからの XML ビュードキュメントの作成

XML ビュードキュメントを作成するには、Model Explorer で要素を選択し、**Modeling > Create XML View Documents** と選択します。

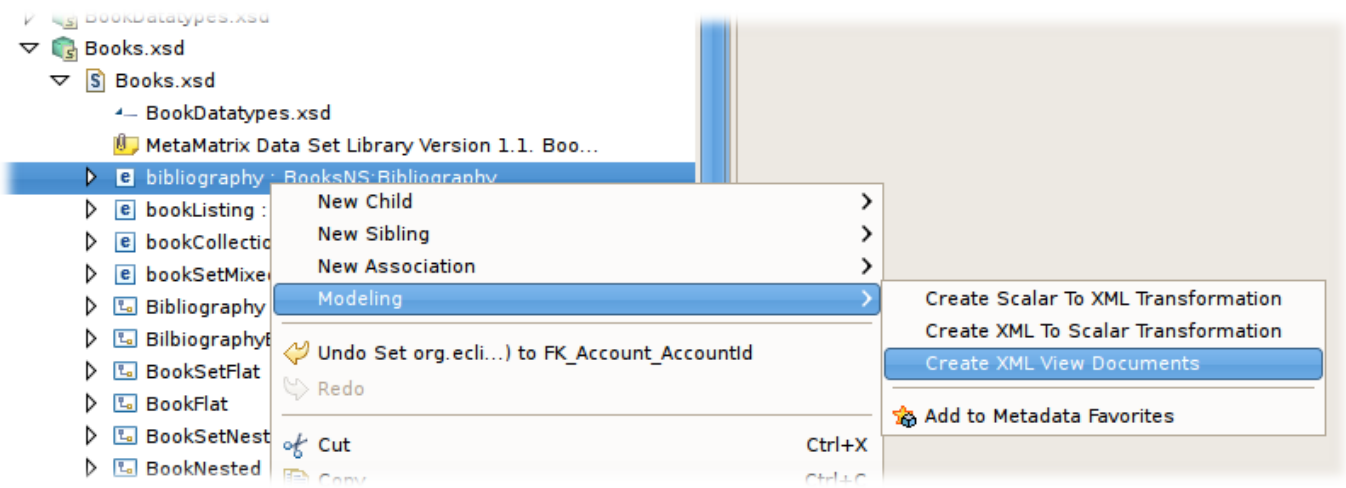


図12.21 XML ビュードキュメントの作成

これにより、目的の XML ドキュメントモデルに対してクエリーを行います。ワークスペースから既存の XML ドキュメントモデルを選択するか、一意のモデル名を入力すると、ウィザードによって新しいモデルが作成されます。

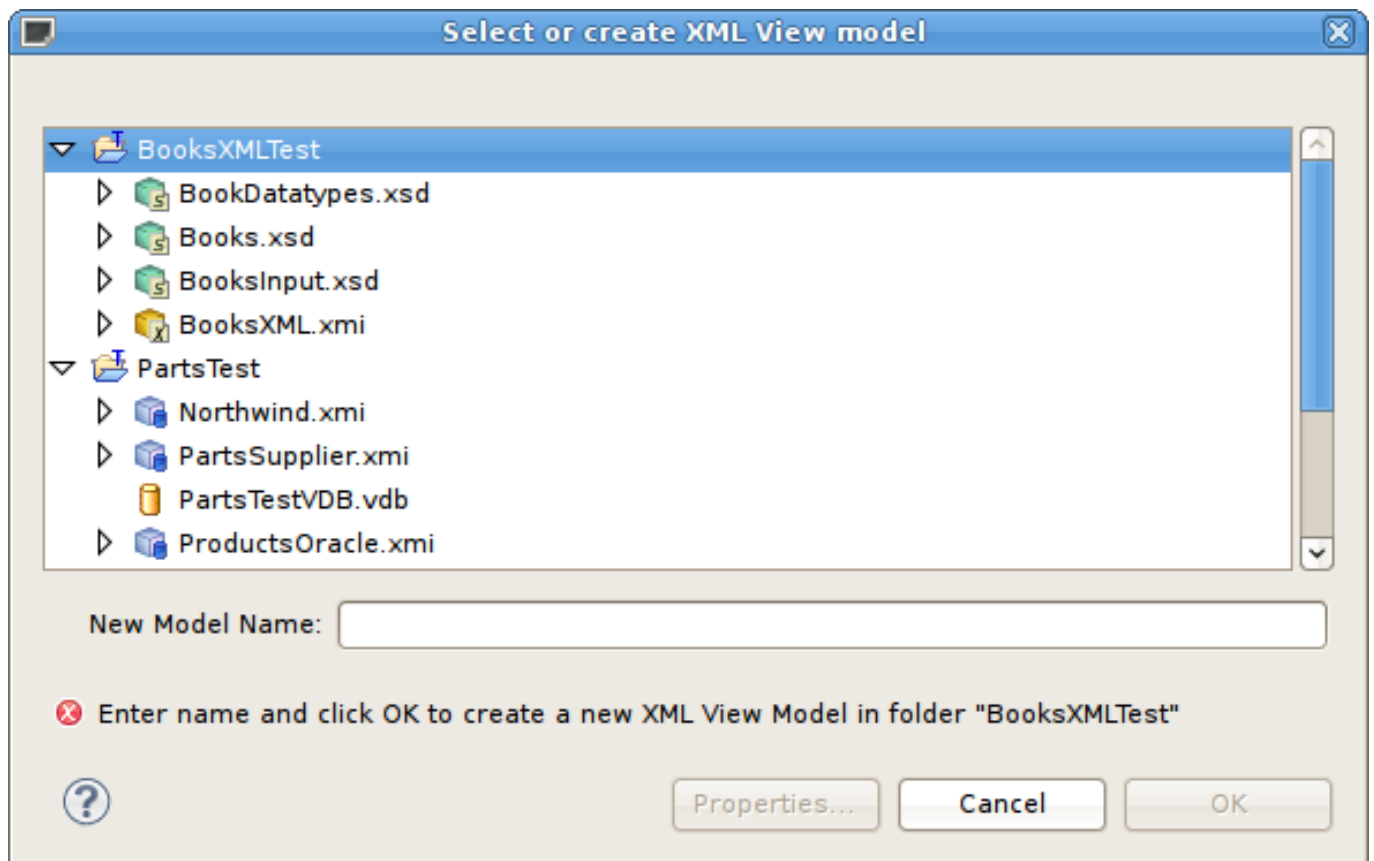


図12.22 Select or Create XML View Model ダイアログ

新しい XML ドキュメントモデルを選択または作成後、XML ドキュメントビルダーページが表示されます。このページは、スキーマセクションからのドキュメントビューで詳細に説明されています。

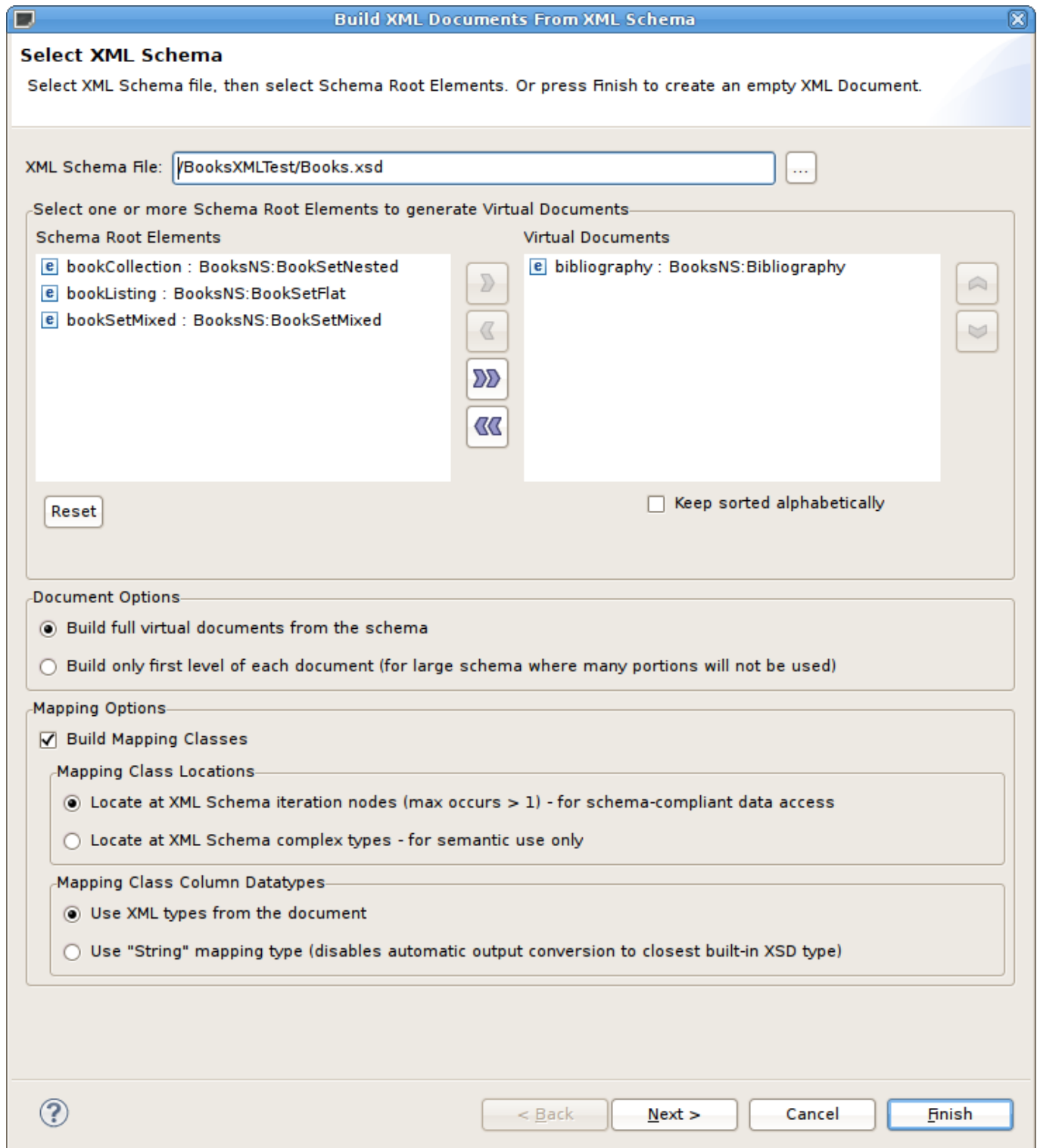


図12.23 Build XML Documents From XML Schema ダイアログ

## 12.4. Web サービスのモデリング

### 12.4.1. Web サービスの作成

このメソッドは、リレーショナルソースのクエリーを目的とする迅速で一貫した Web サービスのデプロイメントを行う経験者に奨励されるメソッドです。XML ビューからのすべての Web サービスインターフェース、演算、および必要なトランスフォーメーションを詳細に制御できます。

リレーショナルモデルまたはオブジェクトから Web サービスモデルを作成するには、以下を実行します。

1. Mode Explorer のビューツリーでリレーショナルモデル、テーブル、プロシーチャーの組み合わせを選択します。



### 注記

手順3で入出力スキーマおよび Web サービスモデルの自動命名が有効になるため、単一のソースモデルを選択することが推奨されます。

2. 右クリックして **Modeling > Create Web Service** アクションを選択します。

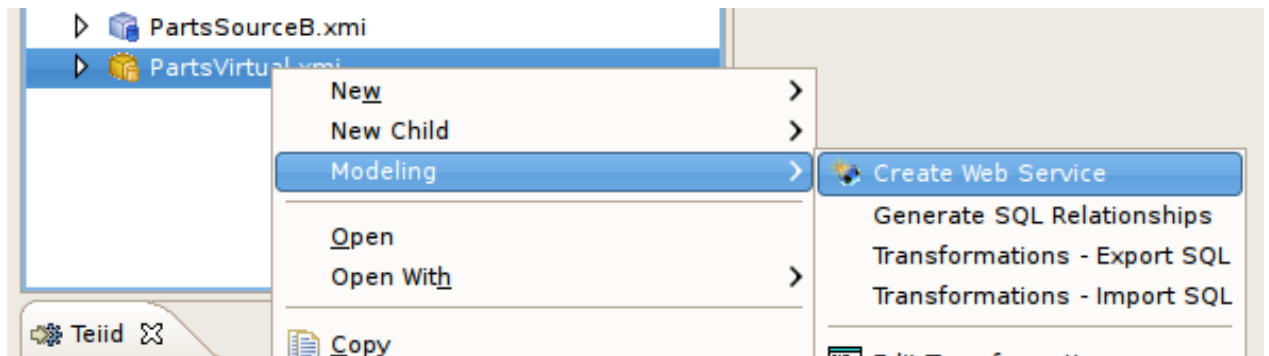


図12.24 Web サービスの作成

3. **Create Web Service** ダイアログで、生成された入力スキーマファイル、出力スキーマファイル、および Web サービスモデルのファイル名を指定します。必要な場合はオプションを変更します。最後に **Finish** をクリックします。

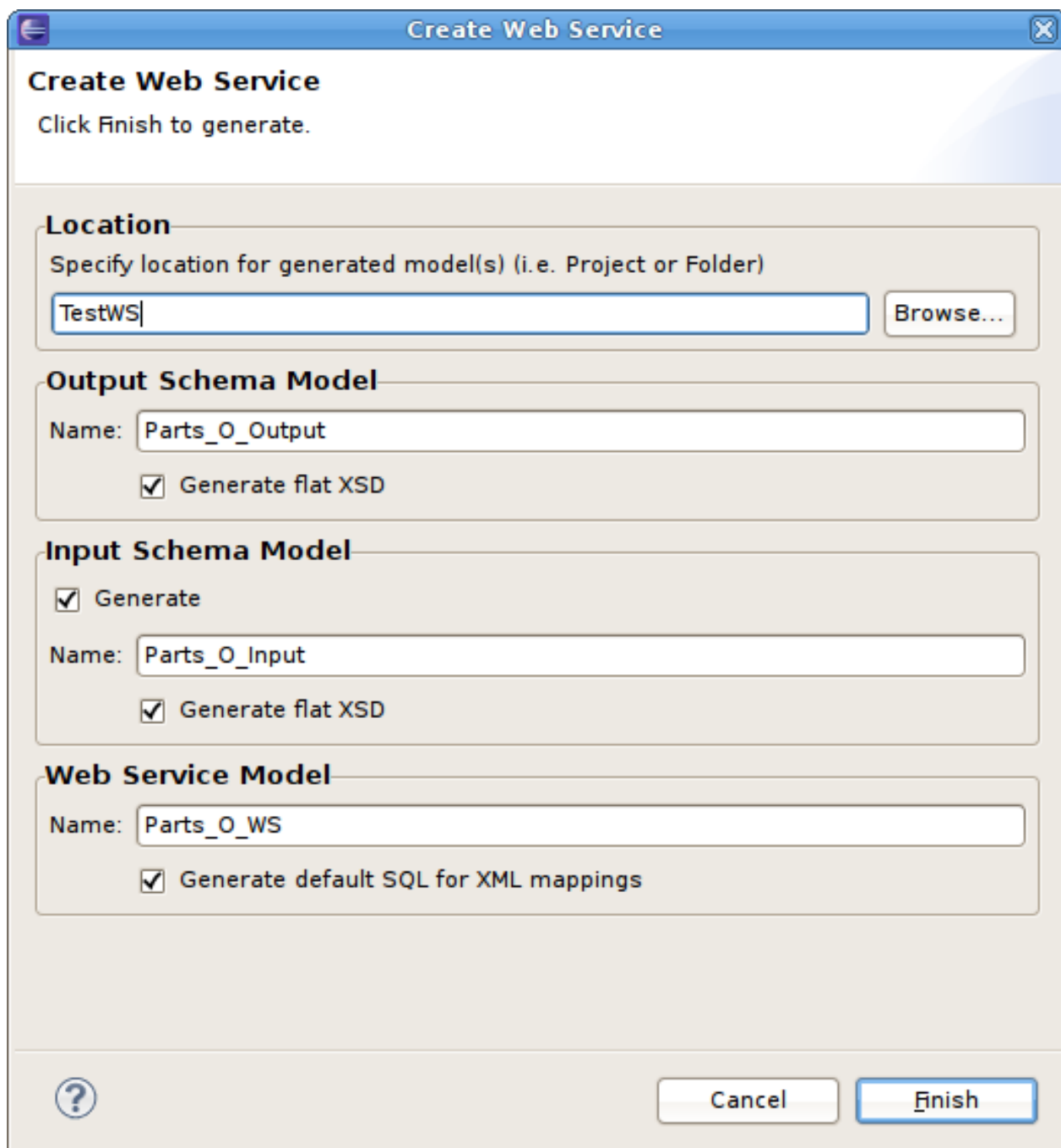


図12.25 Create Web Service ダイアログ

4. モデルの生成が終了したら、確認ダイアログが表示されます。**OK** をクリックします。



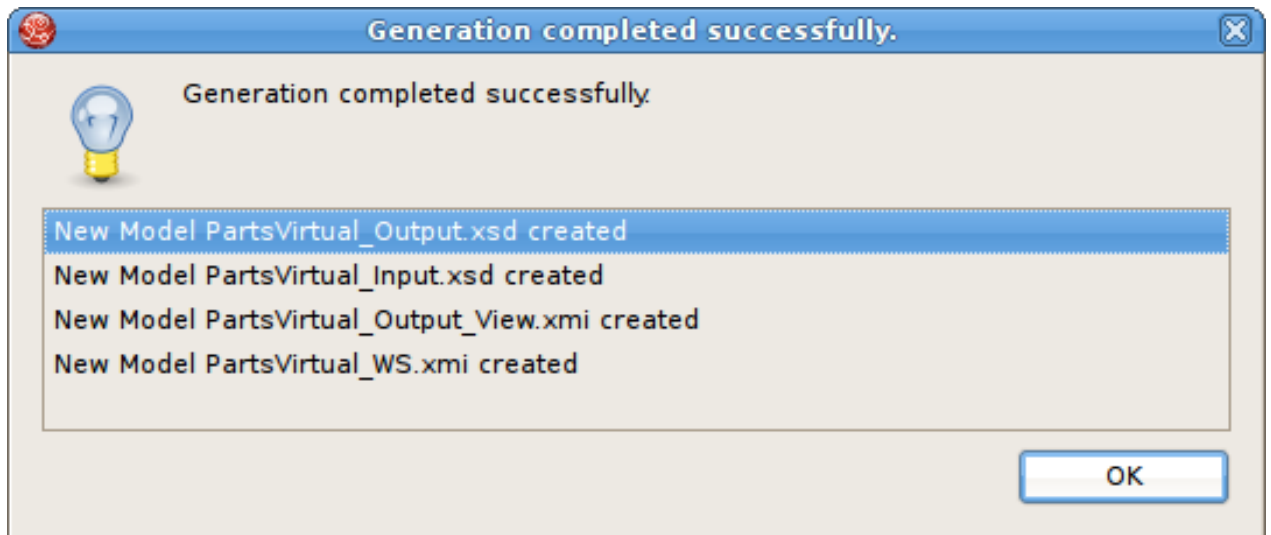


図12.26 Generation Completed ダイアログ

## 12.4.2. Web サービス war の生成

### 12.4.2.1. Web サービス war の生成

**Teiid Designer** では、SOAP または REST インターフェースより VDB を公開できます。JBossWS-CXF または RESTEasy war は、VDB 内のモデルを基に生成できます。

### 12.4.2.2. JBossWS-CXF war の生成

**Teiid Designer** は、JBossWS-CXF war の形式で Web サービス生成機能を提供します。「新しい Web サービスビューモデル」での説明どおり Web サービスモデルを VDB に追加し、VDB を稼働中の JBoss Data Virtualization インスタンスにデプロイし、VDB のデータソースを作成したら、生成された war を使用して Web サービスを公開する準備が整います。

VDB を使用して新しい JBossWS-CXF war を生成するには、以下を行います。

1. Web サービスモデルが含まれる VDB を右クリックし、**Modeling > Generate JBossWS-CXF War** と選択します。

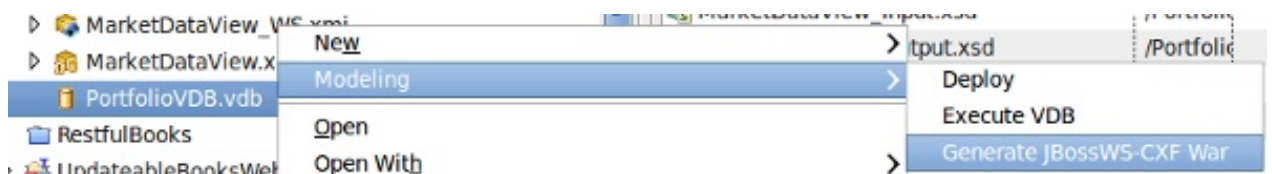


図12.27

2. 以下のように、**Web Service War Generation Wizard** で不足しているプロパティーを入力します。

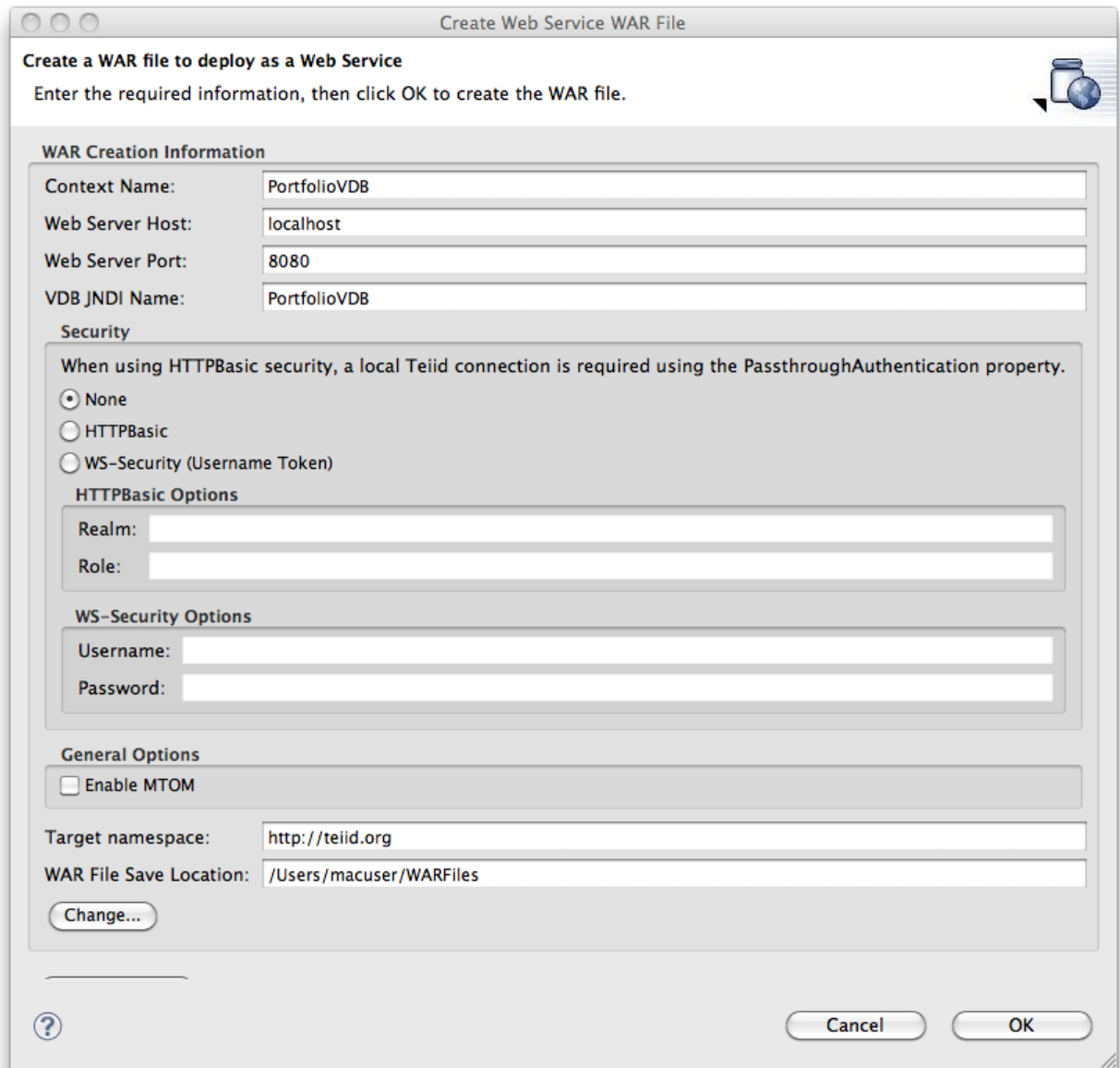


図12.28 Generate a JBossWS-CXF War Web Service ダイアログ

表12.7 フィールドの説明

| フィールド名        | 説明                                 |
|---------------|------------------------------------|
| 名前            | 生成された war ファイルの名前。                 |
| ホスト           | サーバーのホスト名 (または IP)。                |
| ポート           | サーバーポート。                           |
| VDB JNDI Name | デプロイされた Teiid ソース VDB への JNDI 接続名。 |

| フィールド名   | 説明  |
|--|---|
| Security options                                   | <ul style="list-style-type: none"> <li>※ None - 生成された Web サービスを介した VDB への接続にユーザー名やパスワードは必要ありません。</li> <li>※ HTTP Basic - 指定されたセキュリティーレールムおよびロールが使用されます。デフォルトのレールム値は、設定変更のない JBoss Data Virtualization (teiid-security) のレールムです。ロールは適切なセキュリティーメカニズムで定義する必要があります。Teiid の場合、<b>teiid-security-roles.properties</b> ファイルを使用します。HTTPBasic を使用する場合、PassthroughAuthentication プロパティーを使用するローカルの Teiid 接続が必要になります。</li> <li>※ WS-Security - war generator ダイアログに指定したユーザー名とパスワードの値が渡されるか検証するパスワードコールバッククラスが生成されます。これは、WS-Security が有効になっている Web サービスのテストメカニズムと独自のセキュリティーメカニズムがこのクラスに実装される必要があることを意味します。すべてのソースコードがコンパイルされたクラスファイルとともに生成された war に含まれます。</li> </ul> |
| Target namespace                                   | 生成された WSDL および後続の生成された Web サービスクラスで使用される目的の名前空間。  |
| MTOM (Message Transmission Optimization Mechanism) | 生成されると、MTOM は Web サービスエンドポイントに対して有効になります。また、 <code>xmlns:xmime="http://www.w3.org/2005/05/xmlmime" スキーマと <code>type="xs:base64Binary"xmime:expectedContentTypes="application/octet-stream" を最適化する出力要素に追加し、出力スキーマを更新する必要があります。</code></code>   |
| War File Save Location                             | 生成された WAR ファイルが保存されるフォルダー。  |

3. **OK** をクリックし、Web サービス war を生成します。war の生成が完了したら、確認ダイアログが表示されます。**OK** をクリックします。

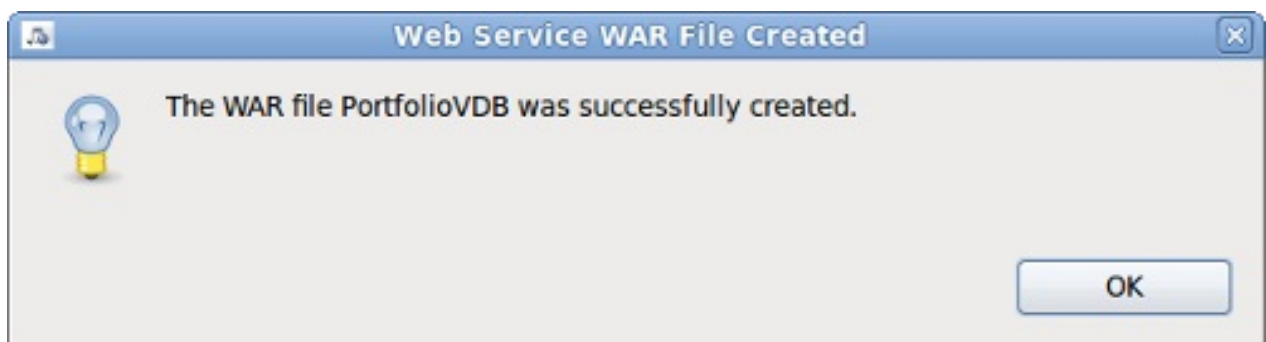


図12.29 Generation Completed ダイアログ

4. **OK** をクリックします。

### 12.4.2.3. RESTEasy war の生成

**Teiid Designer** では、生成された RESTEasy war を使用して REST 上で VDB を公開することも可能です。目的の仮想モデルに SQL の更新、挿入、および削除が定義されている場合、REST より簡単に CRUD 機能を提供できます。生成された REST 演算へ許可される入力は、URI パスパラメーター、クエリーパラメーター、XML/JSON です。JSON は、「json」が含まれる URI 上で公開されます。たとえば、`http://{host}:{port}/{war_context}/{model_name}/resource` は URI パスパラメーターや XML を許可し、`http://{host}:{port}/{war_context}/{model_name}/json/resource` は URI パスパラメーターや JSON を許可します。また、`REST:URI = authors&parm1&parm2` のように & を区切り文字として使用して目的の REST プロシーチャーの URI プロパティにあるクエリーパラメーターを指定できます。

1. 仮想モデルで、XMLLiteral オブジェクトを返すプロシーチャーを追加します。VDB のすべてのモデルをプロシーチャーのターゲットとすることができます。仮想テーブル (VirtualBooks) から選択し、結果を XMLLiteral として返すプロシーチャーの例を以下に示します。

```

CREATE VIRTUAL PROCEDURE
BEGIN
  SELECT XMLELEMENT(NAME books, XMLAGG(XMLELEMENT(NAME book, XMLFOREST(RestfulBooks.VirtualBooks.ISBN, RestfulBooks.VirtualBooks.TITLE, RestfulBooks.VirtualBooks.SUBTITLE, RestfulBooks.VirtualBooks.PUBLISHER, RestfulBooks.VirtualBooks.PUBLISH_YEAR, RestfulBooks.VirtualBooks.EDITION, RestfulBooks.VirtualBooks.TYPE, RestfulBooks.VirtualBooks.AUTHOR_ID)))) AS result FROM RestfulBooks.VirtualBooks WHERE RestfulBooks.rgetBooks.isbn_in = RestfulBooks.VirtualBooks.ISBN;
END

```

図12.30

選択のリレーショナルテーブルの結果を VirtualBooks から XMLLiteral に変換する構文に注目してください。

行を追加し、XMLLiteral オブジェクトを返す更新プロシーチャーの例は次のとおりです。

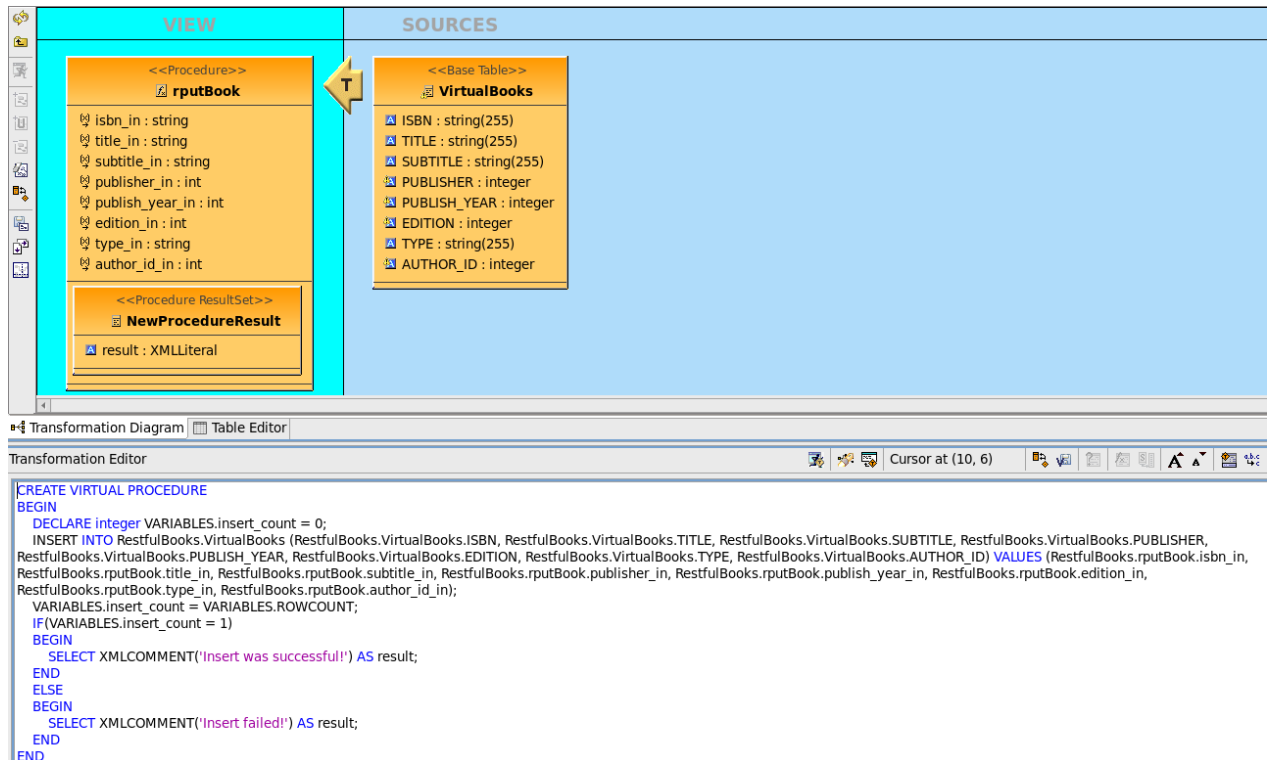


図12.31

REST プロシージャの入力形式は、URI パラメーター、XML/JSON ドキュメント、または両方の組み合わせの一部になります。XML ドキュメントを使用する場合は、ルートノードが <input> である必要があります、XML ノードがプロシージャの入力パラメーターの順番に対応する必要があります。たとえば、上記の挿入プロシージャの入力は次のとおりです。

```

<input>
<ISBN>0-13-014714-1-99999</ISBN>
<TITLE>The XML Handbook</TITLE>
<SUBTITLE>Updated Edition</SUBTITLE>
<PUBLISHER>16</PUBLISHER>
<PUBLISH_YEAR>2000</PUBLISH_YEAR>
<EDITION>2</EDITION>
<TYPE>Hardback</TYPE>
<AUTHOR_ID>49</AUTHOR_ID>
</input>

```

図12.32 XML 入力の例

JSON ドキュメントを使用する場合は、値がプロシージャ入力パラメーターの順番と一致するようになっています。上記の挿入プロシージャの入力は次のとおりです。

```
{
  "ISBN": "1-55615-484-4",
  "TITLE": "Code Complete",
  "SUBTITLE": "A Practical Handbook of Software Construction",
  "PUBLISHER": 5,
  "PUBLISH_YEAR": 1993,
  "EDITION": 1,
  "TYPE": "Hardback",
  "AUTHOR_ID": 31
}
```

図12.33 JSON 入力の場合

- ここでプロシージャを REST 対応であると特定する必要があります。これには、Modeling > Enable REST for Procedures とコンテキストメニューオプションを選択し、プロシージャのプロパティを追加します。

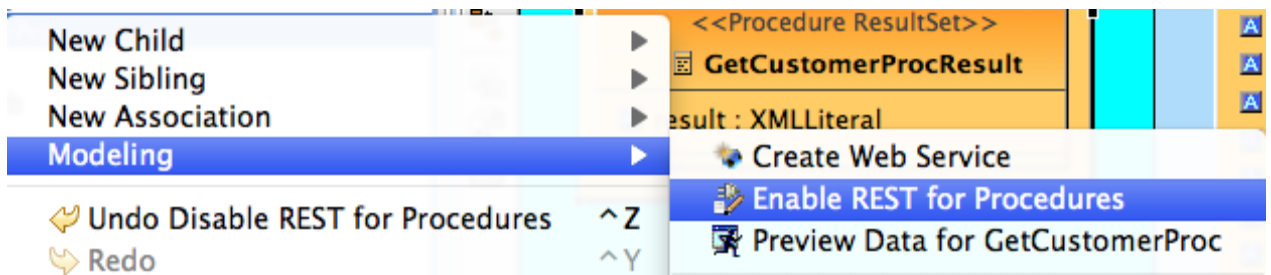


図12.34

これにより、モデルで定義されたすべてのプロシージャに対し、プロパティタブで2つの新しいプロパティが有効になります。必要な2つのプロパティは次のとおりです。

表12.8 RESTful プロシージャの必要な拡張プロパティ

| プロパティ名    | 説明   |
|-----------|--|
| REST メソッド | このプロシージャの REST マッピングを決定する HTTP メソッド。サポートされるメソッドは GET、PUT、POST、および DELETE です。   |
| URI       | プロシージャへのリソースパス。たとえば、books/{isbn} をプロシージャの URI 値として使用する場合、このプロシージャは http://{host}:{port}/{war_context}/{model_name}/books/123 によって実行され、123 がパラメーターとして渡されます。 |

上記の例は Property タブでは次のように表示されます。

| Property         | Value        |
|------------------|--------------|
| ▼Extension       |              |
| rest:Rest Method | GET          |
| rest:URI         | books/{isbn} |

図12.35

生成された URI にはパスの一部としてモデル名が含まれるため、完全な URL は「http://{host}:{port}/{war\_context}/{model\_name}/books/123」のようになります。REST サービスがすべて本 (book) を返すようにするには、上記のとおりにプロシーチャーを記述して入力パラメーターのみ削除します。URI プロパティは「books」のみになり (または指定するもの)、URL は「http://{host}:{port}/{war\_context}/{model\_name}/books」になります。

必要な拡張プロパティとともにすべてのプロシーチャーを追加したら、モデルを VDB に追加します。モデルがすでに VDB に含まれている場合は同期します。その後、VDB を再デプロイする必要があります。



### 重要

開発中に VDB を再デプロイした場合、プールに対して陳腐化した接続が取得され、「Invalid Session Exception」(無効なセッション例外) が発生することがあります。これを修正するにはデータソースをフラッシュしますが、テストクエリーを VDB 接続の -ds.xml ファイルに追加して修正することもできます。これにより、VDB を再デプロイした後に確実に有効な接続を取得することができます。テストクエリーの構文は `<check-valid-connection-sql>some arbitrary sql</check-valid-connection-sql>` になります。

3. VDB のデータソースを作成していない場合は作成する必要があります。これは、Teiid Designer の **Server view** で行うことができます。デプロイした VDB を右クリックし、**Create Data Source** を選択します。VDB に接続するため、Generate REST WAR ダイアログで作成したソースの JNDI 名が要求されます。
4. REST 対応のプロシーチャーを持つ仮想モデルが含まれる VDB を右クリックし、Modeling > Generate RESTEasy War と選択します。REST 対応のプロシーチャーがない場合、「Generate RESTEasy War」オプションは有効になりません。

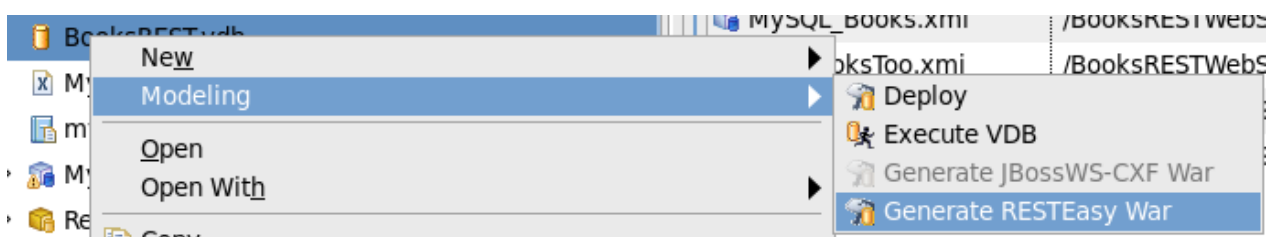


図12.36

5. 以下のように **REST War Generation Wizard** で不足しているプロパティを入力します。

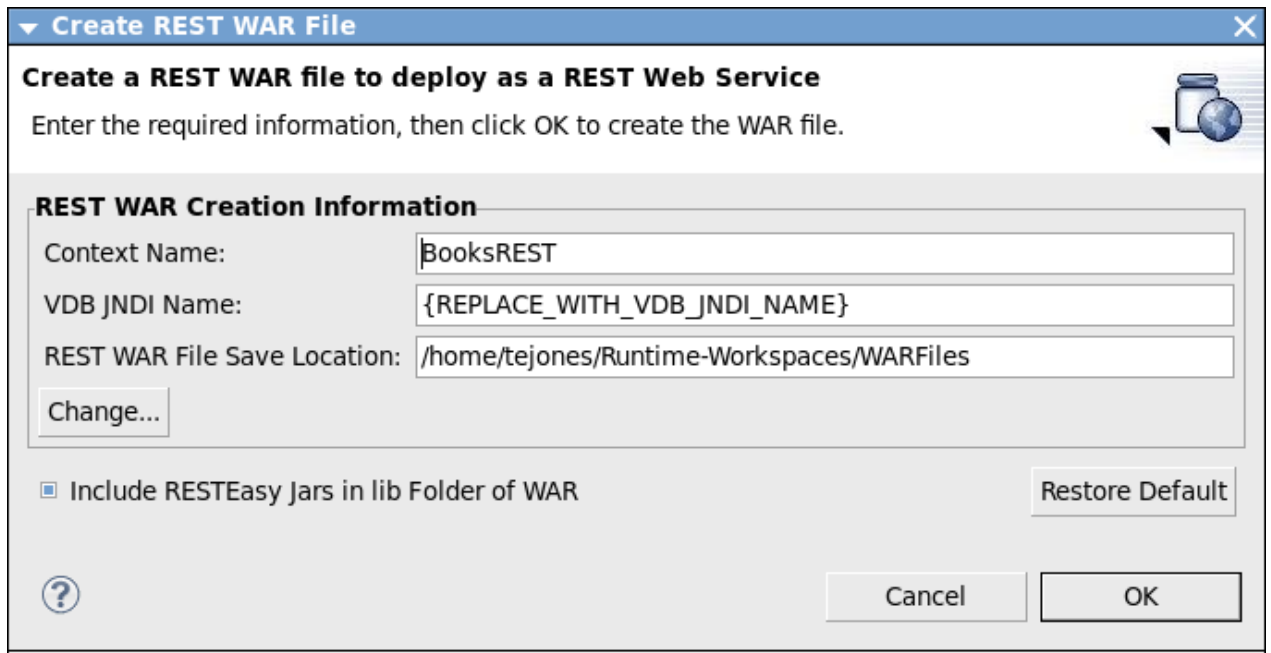


図12.37 Generate a REST WAR War File ダイアログ

表12.9 フィールドの説明

| フィールド名                                     | 説明   |
|--|--|
| 名前   | 生成された war ファイルの名前。   |
| Connection JNDI Name                       | デプロイされた Teiid ソース VDB への JNDI 接続名。   |
| War File Save Location                     | 生成された WAR ファイルが保存されるフォルダー。   |
| Include RESTEasy Jars in lib Folder of WAR | 選択すると、RESTEasy jar とそれらが依存する jar が生成された WAR の lib フォルダーに含まれるようになります。選択しないと、jar は含まれません。RESTEasy がサーバーインストールのクラスパスにインストールされている環境では、これを未選択にして競合を防ぐ必要があります。 |

6. **OK** をクリックして REST war を生成します。war の生成が完了したら確認ダイアログが表示されず。**OK**をクリックします。

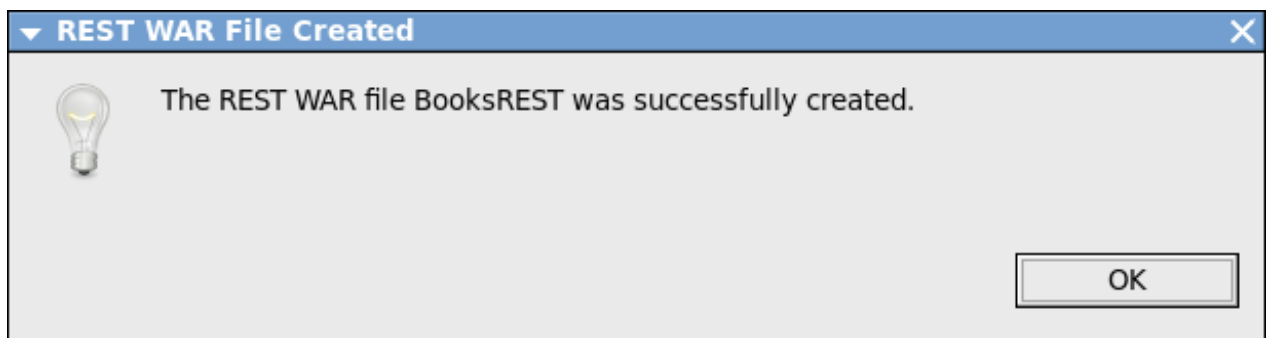


図12.38 Generation Completed ダイアログ

#### 12.4.2.4. 生成された WAR ファイルのデプロイ



war ファイルを生成したら、JBoss AS インスタンスにデプロイする必要があります。デプロイを行う方法は複数あります。

### JBDS または JBoss Tools から

1. 目的の JBoss AS が設定され、実行されていることを確認します。
2. Model Explorer ビューで WAR ファイルを選択します。その場所に war を生成しなかった場合は、コピーして貼り付けできます。
3. WAR ファイルを右クリックし、**Mark as Deployable** 選択します。これにより、定義した JBoss AS インスタンスに WAR ファイルが自動的にデプロイされます。

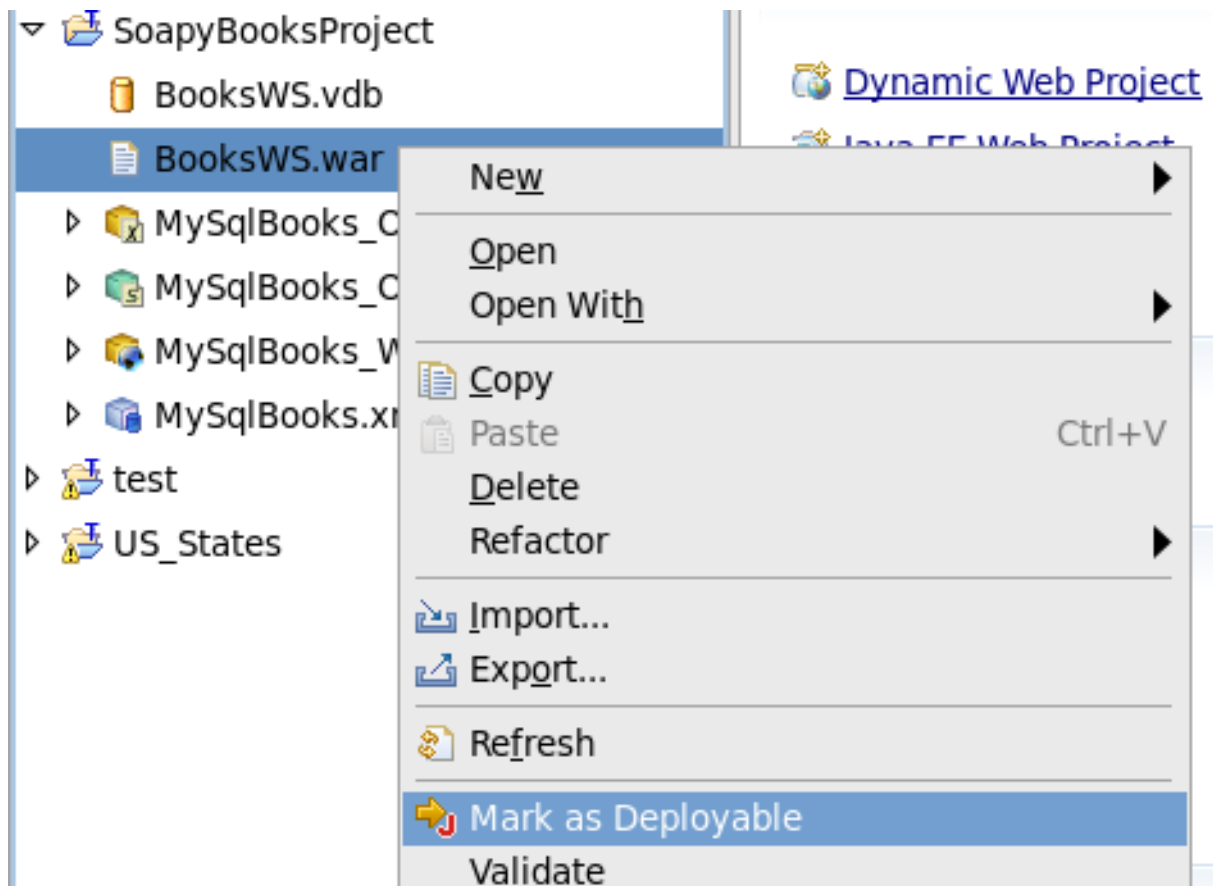


図12.39

### JBoss AS 管理コンソールを使用

1. JBoss AS に同梱される管理コンソールを使用して、WAR ファイルをデプロイできます。管理コンソールは、`http://{host:port}/admin-console` にあります。ログインしたら、Web Application (WAR) リソースフォルダーの **Add a New Resource** ボタンを使用します。

### JBoss AS への手動デプロイメント

1. 手動でファイルを JBoss AS の `deploy` フォルダーにコピーして、生成された WAR をデプロイすることも可能です。サーバーが稼働中の場合は、ホットデプロイで WAR は自動的にデプロイされます。それ以外の場合は、サーバーが次回起動したときに WAR がデプロイされます。

#### 12.4.2.5. 生成された WAR ファイルのテスト

war ファイルをデプロイしたら、テストすることができます。テストする方法は複数あります。

## SOAP WAR テスト

### WSDL URL の判断

http://{server:port}/jbossws/services で WSDL URL を確認できます。ここでは、目的の JBoss AS サーバーのデプロイされた web サービスがすべてリストされます。サービスを見つけ、**Endpoint Address** リンクをクリックします。これにより、web サービスの WSDL が取得され、WSDL URL アドレスがブラウザのアドレスバーに表示されます。

```
Endpoint Name    jboss.ws:context=books,endpoint=MySqlBooks_BOOKS
Endpoint Address http://127.0.0.1:8080/books/MySqlBooks_BOOKS
```

| StartTime                    | StopTime          |                   |
|------------------------------|-------------------|-------------------|
| Mon Jun 11 15:16:02 CDT 2012 |                   |                   |
| RequestCount                 | ResponseCount     | FaultCount        |
| 0                            | 0                 | 0                 |
| MinProcessingTime            | MaxProcessingTime | AvgProcessingTime |
| 0                            | 0                 | 0                 |

図12.40

WSDL URL を取得すると、JBDS や JBoss Tools に同梱される Web Service Tester や soapUI などの外部ツールを含むすべての SOAP テストツールを使用できます。

### JBoss AS 管理コンソールを使用

JBoss AS に同梱される管理コンソールを使用して、WAR ファイルをデプロイできます。管理コンソールは、http://{host:port}/admin-console にあります。ログインしたら、Web Application (WAR) リソースフォルダーの Add a New Resource ボタンを使用します。

## REST WAR テスト

### URI

REST プロシーチャーをモデル化したとき、対応する演算 (GET、PUT、POST、または DELETE) とともに定義された各 HTTP 演算に対して URI を割り当てました。各 URI のフルパスは、XML 入力/出力の場合は /{war\_context}/{model\_name}/{resource} として定義され、JSON 入力/出力の場合は /{war\_context}/{model\_name}/json/{resource} として定義されます。

WSDL URL を使用すると、JBDS や JBoss Tools に同梱される Web Service Tester、および soapUI や cURL などの外部ツールなど、REST サポートが含まれるすべてのテストツールを使用できます。

## 第13章 モデルおよびプロジェクトの編集

### 13.1. モデルおよびプロジェクトの編集

**Teiid Designer** では、Rename (名前変更)、Move (移動)、および Save As... (別名で保存) の 3 つの基本的なモデル変更アクションと、プロジェクトに関連する Clone Project (プロジェクトのクローン) が提供されます。

### 13.2. モデル名の変更

ワークスペースのモデルの名前を変更するには、以下を実行します。

1. **Model Explorer** ビューでモデルを選択します。
2. 右クリックして **Refactor > Rename** と選択します。

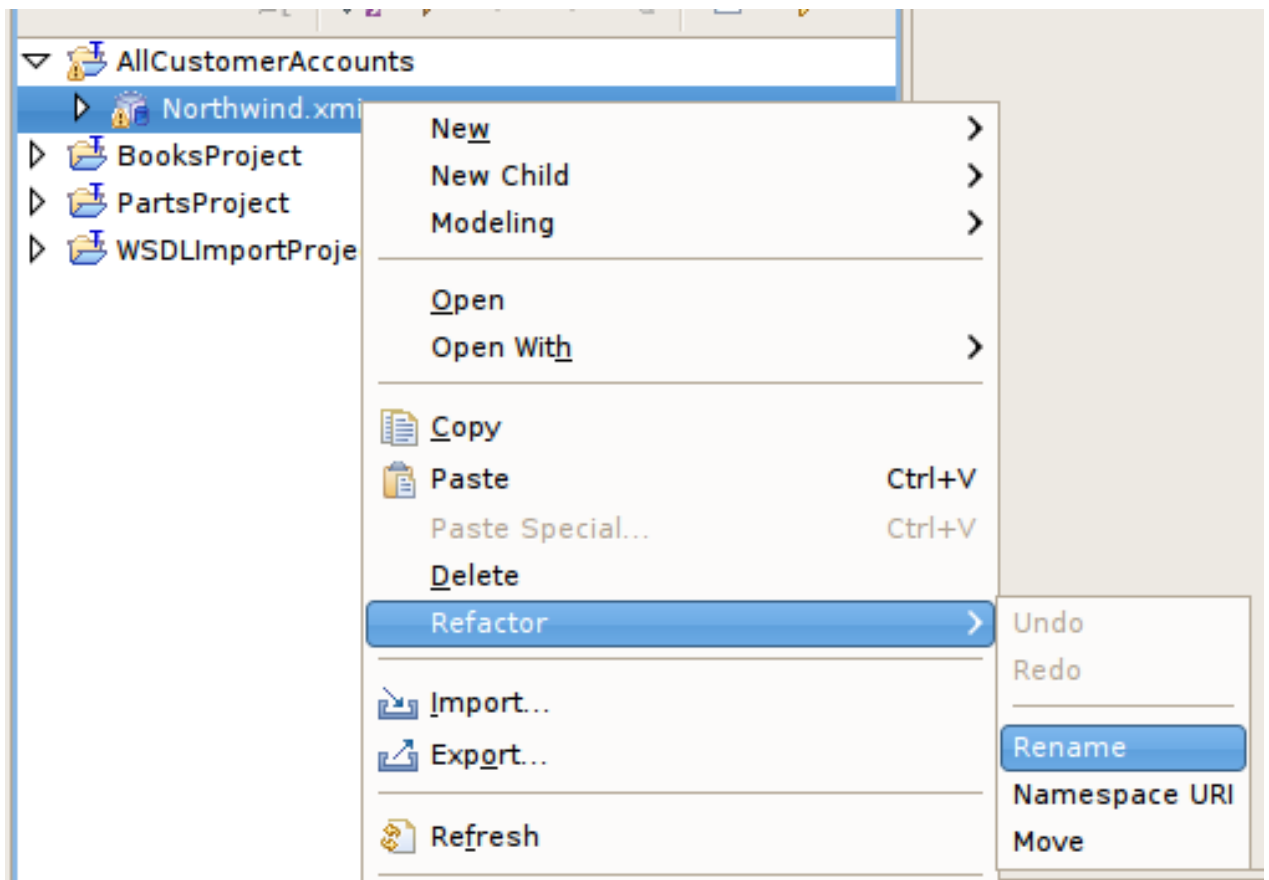


図13.1 Model Explorer の Refactor - Rename アクション

3. **Rename Model File** ダイアログで一意的モデル名を指定します。OK をクリックします。

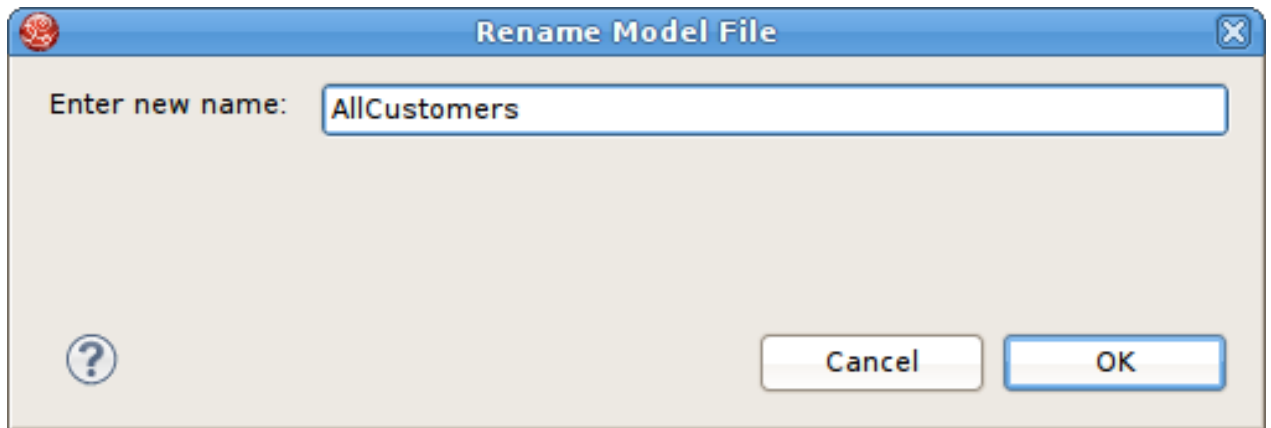


図13.2 Rename Model File ダイアログ



### 注記

別のモデルに依存するモデルの名前を変更すると、モデルのモデルインポートが自動的に変更されます。たとえば、ソースモデル CustomerSource が OldCustomerSource という名前に変更されると、CustomerSource をインポートするビューモデル CustomerAccounts のインポートステートメントが新しい名前を反映して変更されます。

## 13.3. モデルの移動

ワークスペースのモデルを移動するには、以下を実行します。

1. **Model Explorer** ビューでモデルを選択します。
2. 右クリックして **Refactor > Move** と選択します。

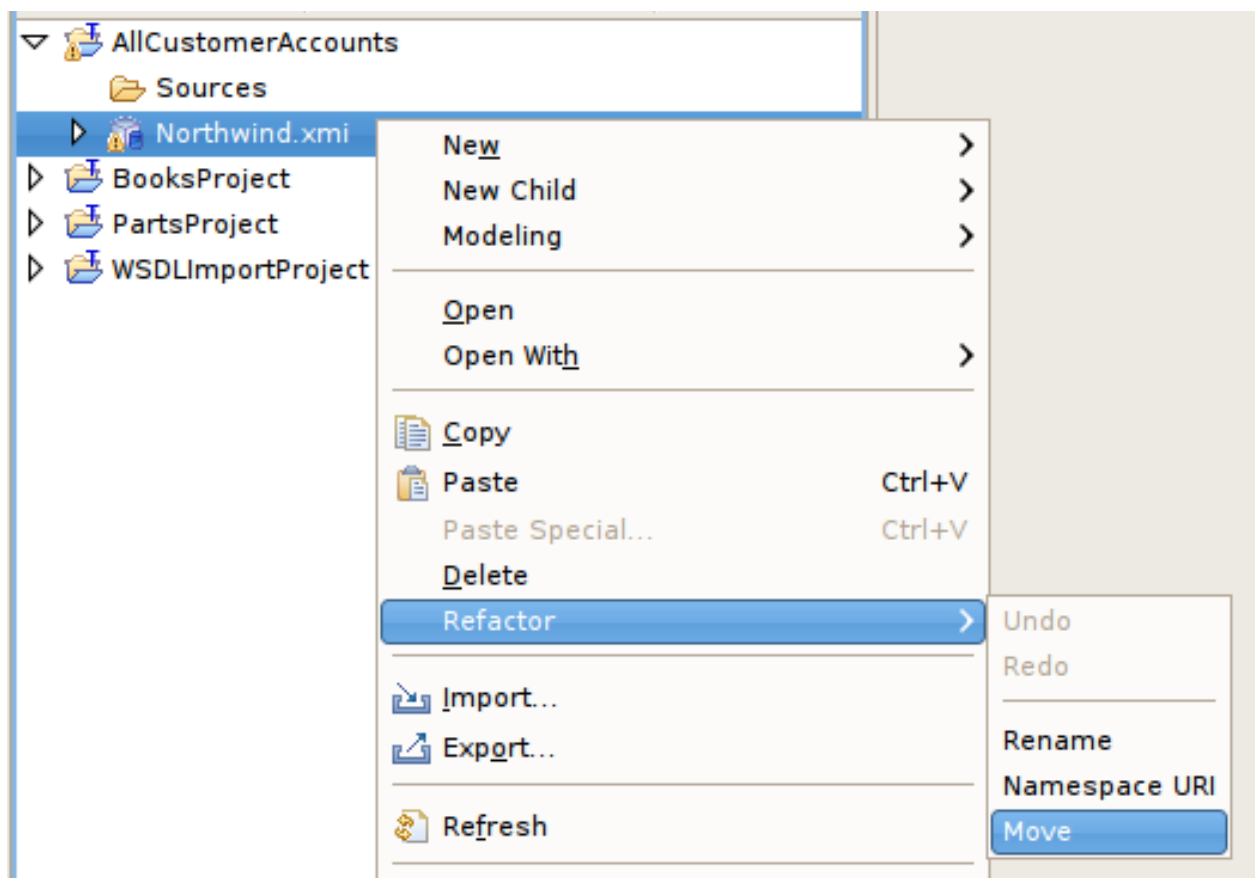


図13.3 Model Explorer の Refactor - Move アクション

3. 新しいフォルダーの場所を選択し、**OK** をクリックします。

## 13.4. モデルのコピーの保存

**Save As...** は **Refactor > Rename** と同様の機能を実行しますが、名前を変更したモデルは元のモデルの構造的なコピーになります。

### 注記

各モデルオブジェクトは独自の一意な ID を保持するため、モデルのコピーは構造が元のモデルと同じになりますが、一意なオブジェクト ID が再生成されます。ローカルファイルシステムでモデルを見つけて移動すると、**Teiid Designer** 内ではランタイムエラーが発生する可能性があることに注意してください。各モデルは一意であることが想定され、複製のモデルは許可されません。

「Save As...」を使用して複製のモデルを作成するには、以下を行います。

1. Model Explorer ビューでモデルをダブルクリックするか、右クリックして **Open** を選択し、Model Editor でコピーするモデルを開きます。
2. 開いたモデルの Editor タブを選択します。

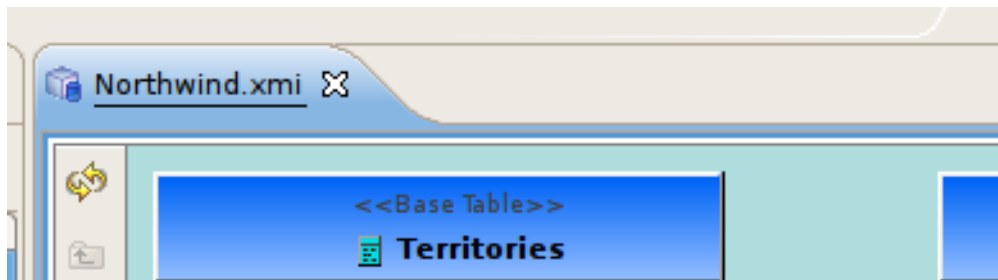


図13.4 Editor タブの選択

3. **File > Save As...** と選択し、**Save Model As** ダイアログを開きます。

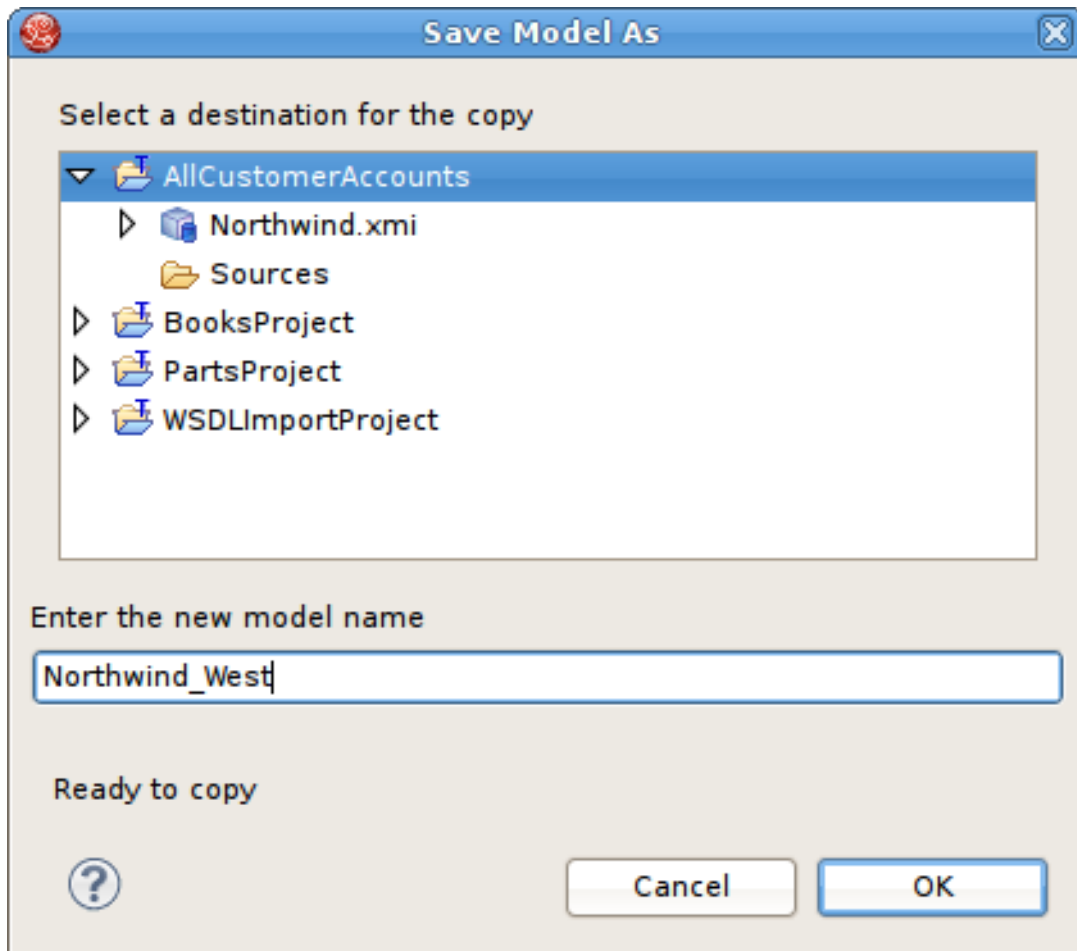


図13.5 Save Model As ダイアログ

4. new model name テキストフィールドに一意なモデル名を入力し、**OK** をクリックします。
5. 依存するモデルが検出された場合、**Save Model As - Import References** ダイアログが表示され、依存するモデルのインポートが新しいモデルを参照するかどうかを変更できます。

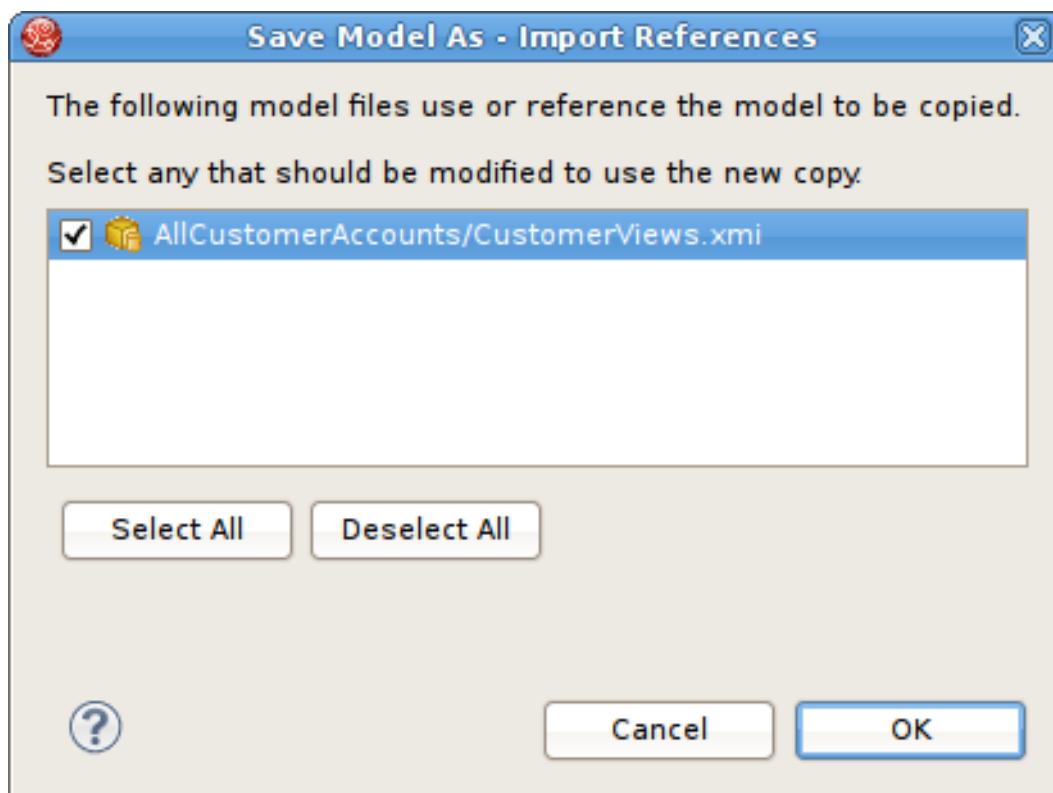


図13.6 Save Model As ダイアログ

## 13.5. プロジェクトのクローン

モデルの各インスタンスには一意な ID が含まれ、各モデルの各オブジェクトには一意な ID が含まれるため、プロジェクトをコピーするときは注意が必要です。そのため、ソースプロジェクトのモデルすべての構造的なコピーを作成し、管理するため **Clone Project** アクションが作成されました。

このアクションのルールや制限は次のとおりです。

- ※ このアクションは、ユーザー定義のディレクトリー構造で組織されたモデルファイル (XMI または XSD) が任意の数含まれる、完全なモデルプロジェクトをクローンします。
- ※ 元のオブジェクト内のすべてのオブジェクト参照 (UUID) は新しい一意な参照に置き換えられます。
- ※ モデルの依存関係または内部オブジェクト参照はクローンされたプロジェクト内の依存関係を反映するためにリファクタリングされます。
- ※ 元のプロジェクトの外部にあるプロジェクトのモデルへのモデル参照は置き換えられません。
- ※ XMI および XSD ファイルのみがクローンされます。プロジェクトの他のファイルタイプはすべて処理されず、VDB を含む新たにクローンされたプロジェクトにコピーされません。
- ※ 保存が必要な 1 つまたは複数のエディターが開かれている場合、クローン処理を継続する前に保存するよう要求されます。

モデルプロジェクトをクローンするには、以下を実行します。

1. **Model Explorer** ビューで既存のモデルプロジェクトを選択します。
2. コンテキストメニューで右クリックして **Model Project > Clone** と選択します。または、**Teiid Designer** のメインメニューバーにある **Project > Clone Project** アクションを選択します。

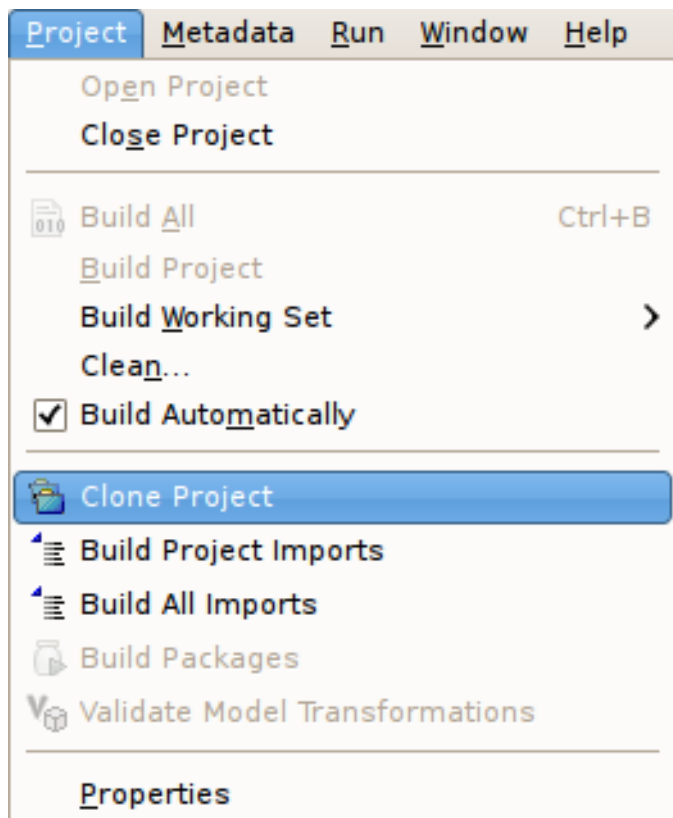


図13.7 Project メニューの Clone Project

3. **Clone Project** ウィザードページで、新しいプロジェクトの名前を指定します。



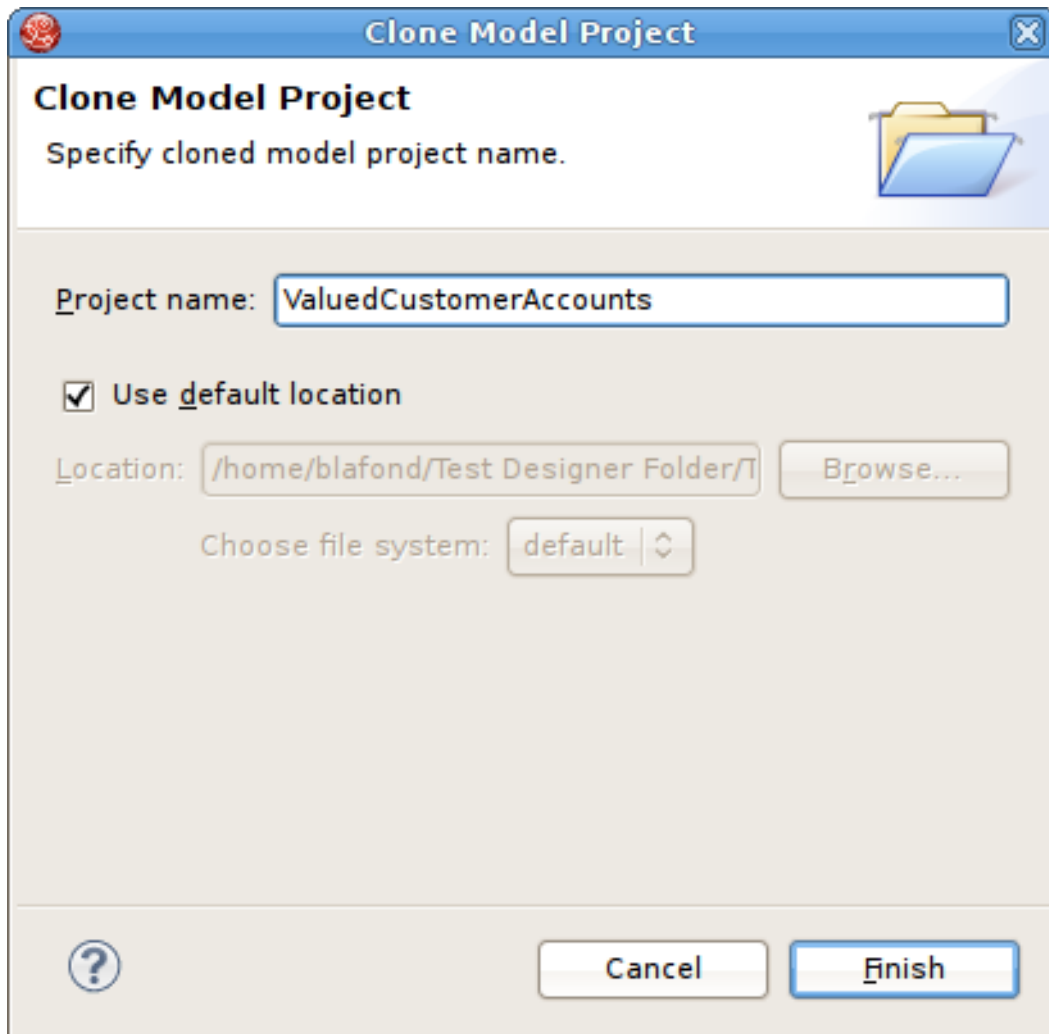


図13.8 Project メニューの Clone Project

4. (任意設定) デフォルトのワークスペース以外の場所にクローンされたプロジェクトを作成する場合は、**Use default location** チェックボックスのチェックマークを外し、ローカルファイルシステムの新しいディレクトリーの場所を指定します (直接入力するか Browse ボタンを押して選択します)。
5. **Finish** をクリックして新しいプロジェクトを生成します。

## 第14章 仮想データベースの管理

### 14.1. 仮想データベースの作成

#### 手順14.1 仮想データベースの作成

1. Red Hat JBoss Developer Studio の **New** ウィザードを起動します。
2. **Teiid Designer** カテゴリディレクトリーを開き、**Teiid VDB** を選択します。



#### 注記

また、モデルプロジェクトで1つ以上のモデルを選択できます。モデルを選択するには、右クリックして **New - Teiid VDB action** と選択します。

このウィザードを起動すると、新しい VDB ダイアログが開かれます。選択したモデルを1つまたは複数起動した場合は、新しい VDB に含まれる事前設定されたモデルがダイアログに表示されません。



#### 注記

VDB は、同じモデルプロジェクト内のモデルやファイルを VDB として認識するようスコープ付けされます。異なるプロジェクトに存在する VDB へモデルを追加することはできません。

### 14.2. 仮想データベースの編集

#### 手順14.2 既存の仮想データベースの編集

1. **Explorer** で編集する VDB を選択します。
2. 右クリックして **Open** を選択します (ダブルクリックすることもできます)。

エディターで VDB が開かれます。

### 14.3. 複数ソースのバインディングサポート

Teiid Designer では、リレーショナルソースモデルを定義し、複数のデータソースにバインドすることができます。

複数ソースのモデルを使用して均一なメタデータを持つ複数のソースのデータに即座にアクセスします。同じスキーマを使用する複数のインスタンスがある場合、Red Hat JBoss Data Virtualization では「マルチソース」モデルを使用してすべてのインスタンス全体でデータを収集できるようにします。この場合、各データソースのモデルを作成またはインポートする代わりに、スキーマを表す1つのソースモデルが定義され、その下にある複数データ「ソース」を用いて設定されます。ランタイム中にこのモデルに対してクエリーが発行されると、クエリーエンジンは情報を分析し、設定したすべてのソースから必要なデータを収集します。さらに、結果を収集して単一の結果を提供します。すべてのソースが同じ物理メタデータを使用するため、この機能は複数のインスタンスを持つ同じソースタイプにアクセスする場合に最も適しています。

VDB エディターの **Models** タブには、左側に簡素化されたモデルテーブル、右側に **Model Details** および **Source Binding Definition** タブが含まれる新しいタブ付きのパネルが含まれます。ソースバインディングを追加するには **Multi-source** チェックボックスをクリックします。



### 注記

各バインディングは、一意のソース名と、サーバーのデプロイされたデータソースを表す一意の JNDI 名を使用して定義する必要があります。

## 14.4. ユーザー定義関数のサポート

Teiid Designer では、ユーザー定義関数 (UDF: User Defined Function) を作成、管理、および使用することができます。これらの関数を使用すると、ランタイム中にデータで簡単または複雑な Java 関数を実行することができます。これを実行するには、サーバーにカスタムの UDF jar をデプロイし、ビュートランスフォーメーションで使用する関数メソッドを表すスカラー関数を作成します。VDB エディターでは、任意で UDF を VDB アーティファクトの一部として含めることができます。jar が VDB に含まれると、VDB がデプロイされたときに jar が自動的にサーバーへデプロイされます。

UDF モデルが VDB に追加されると、各スカラー関数が問い合わせされ、エディターの **UDF Jars** タブに表示されるように参照された UDF jar (ある場合) も VDB に追加されます。

## 14.5. 仮想データベースの再使用

データベースカテゴリーが VDB 名で、VDB で可視できる各モデルがスキーマとして扱われる場合、デプロイされた VDB を別のデータベースとして扱うことができます。これは、**vdb.xml** 定義の **import-vdb** XML 要素を用いて実現できます。VDB が他の VDB を参照できるようにすることで、ユーザーは再使用可能なデータベースコンポーネントを作成し、複雑なトランスフォーメーションの作成に必要なモデリングの量を減らすことができます。

この **vdb.xml** ファイルの例は、ビューモデルのモデル要素内の **import-vdb** 要素と対応する **import-vdb-reference** を強調しています。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<vdb version="1" name="PartssupplierViewsVDB">
  <property value="false" name="preview"/>
  <import-vdb import-data-policies="false" version="1"
name="PartssupplierSourcesVDB"/>
  <model visible="true" type="VIRTUAL" name="PartsViewModel"
path="/PartssupplierProject/PartsViewModel.xmi">
    <property value="1623826484" name="checksum"/>
    <property value="Relational" name="modelClass"/>
    <property value="false" name="builtIn"/>
    <property value="655076658.INDEX" name="indexName"/>
    <property value="PartssupplierSourcesVDB" name="import-vdb-reference"/>
  </model>
</vdb>
```

Teiid Designer では、ユーザーは JDBC Import オプションを使用してデプロイされた VDB からメタデータをインポートでき、これによってこの機能が実現されます。このインポートにより、仮想データベースのカatalog (VDB)、スキーマ (モデル) およびテーブルを構造的に表すリレーショナル VDB ソースモデルが作成されます。

これらの VDB ソースモデルを扱う場合、以下の制限やルールが適用されます。

- ※ VDB ソースは読み取り専用です。
- ※ VDB ソースモデル名は、インポートされた VDB のデプロイされたモデル名 (スキーマ) によって判断されます。
- ※ モデル名はモデルプロジェクト内で一意である必要があります。
- ※ VDB ソースモデルは、再使用 VDB の作成およびデプロイに使用されたプロジェクトとは異なるプロジェクトでインポートまたは作成される必要があります。
- ※ JDBC インポートウィザードは、これらのルールに準拠するオプションを制限します。

## 14.6. VDB ソースモデルの作成

### 手順14.3 VDB ソースモデルの作成

1. VDB をデプロイします。
2. **Import - Teiid Designer - JDBC Database Source Model** とクリックし、**JDBC Import Wizard** を起動します。
3. ウィザードの最初のページで、デプロイされた VDB の有効な接続プロファイルを選択します。

ウィザードによって接続プロファイルが Teiid VDB 接続であることが検出され、選択が **Teiid VDB Source Options** というタイトルのウィザードページに表示されます。



#### 注記

**Import as VDB source model** にチェックマークが入っていないと、ウィザードは通常の JDBC インポートとしてインポートを続行します。

4. **Select Database Objects** ページで、VDB ソースモデルとして作成するために使用するスキーマを 1 つ選択します。
5. 最後のページには、結果となる VDB ソースモデルの名前が表示されます。名前は変更できません。



#### 注記

他のオプションはすべて無効になります。



#### 注記

目的の Into Folder に同じ名前のモデルが含まれないようにしてください。含まれると Finish ボタンが無効化されます。



## 注記

VDB ソースモデルはプロジェクトの他のソースモデルと同様に使用できます。VDB ソースモデルテーブルはトランスフォーメーションクエリーで使用でき、ビューモデルには VDB ソースモデルへのモデルインポートが含まれます。しかし、ビューモデルが VDB に追加されても、参照された VDB ソースモデルは VDB に追加されません。代わりに、import-vdb 要素参照が追加されます。

VDB の VDB インポートが存在する場合、**Show Import VDBs** ボタンが有効になり、インポートされた VDB の名前が表示されます。

## 14.7. セキュリティーおよびデータアクセス

VDB エディターで VDB のデータアクセスセキュリティーを定義するオプションは複数あります。

最初のレベルは、Models 選択 (Spyglass 列) のモデル可視性のチェックボックスによって提供されます。チェックマークを入れないと、モデルとその内容は標準の JDBC メタデータを持つ Teiid ランタイムによって返されません。

次のレベルのセキュリティーは、VDB Editor の下部パネルで管理できるデータロールのパーミッションを定義するために提供されます。一意のデータロールでは、そのモデル内の各モデルとほとんどのオブジェクトには、以下を含むデータアクセスの特定の値を指定できます。

- ✦ Security (行ベースの状態および列マスキング)
- ✦ 作成
- ✦ Read
- ✦ アップデート
- ✦ 削除
- ✦ Execute
- ✦ Alter

テーブルまたは列の **Security** ボックスをダブルクリックすると、適用可能な値を定義できる **Model Object Security Definition** ダイアログが起動されます。セキュリティーを編集または削除するには、**Conditions** または **Masking** タブを選択し、**Edit** または **Remove** ボタンを使用します。

**Permissions** セクションの **Conditions** および **Masks** タブではモデルオブジェクトのこれらの値を管理 (追加、削除、および編集) できます。

## 第15章 モデルのテスト

### 15.1. 接続プロファイルの管理

#### 15.1.1. 接続プロファイルの管理

As described earlier, you can test your models in **Teiid Designer** by using the **Preview Data** action  or test your models via your deployable VDB.

**Teiid Designer** は接続管理に Eclipse Data Tools Platform (DTP) 接続プロファイルフレームワークを使用します。接続プロファイルは、メタデータソースモデルの構築でメタデータにアクセスするため JDBC および非 JDBC ソースに接続するメカニズムを提供します。また、**Teiid Designer** はデプロイされた VDB への JDBC ソースとする、カスタムの Teiid 接続プロファイルテンプレートも提供します。

さまざまな Teiid Designer のインポートオプションを選択すると、Database Development パースペクティブで定義した該当する接続プロファイルをインポートソースとして使用できるようになります。インポートウィザードでは、新しい接続プロファイルを作成したり、ウィザードを終了せずに既存の接続プロファイルを編集することができます。

Server ビューは、稼働中の Teiid インスタンスへのアクセスを提供し、そこにデプロイされたデータソースおよび VDB アーティファクトを表示します。このビューで使用できる Create Data Source アクションは、使用可能な該当する接続プロファイルを利用します。

#### 15.1.2. ソースモデルの接続プロファイルの設定

**Teiid Designer** は、各ソースモデルの関係する接続情報を永続化してデータツール接続プロファイルを統合します。これはインポートプロセスまたは **Modeling > Set Connection Profile** と選択して実行することができます。

#### 15.1.3. ソースモデルの接続プロファイルの表示

ソースモデルに接続プロファイルを設定する以外に、接続の詳細なプロパティを表示するために **Modeling > View Connection Info** と選択し、ソースモデルの接続プロファイル情報を表示することも可能です。

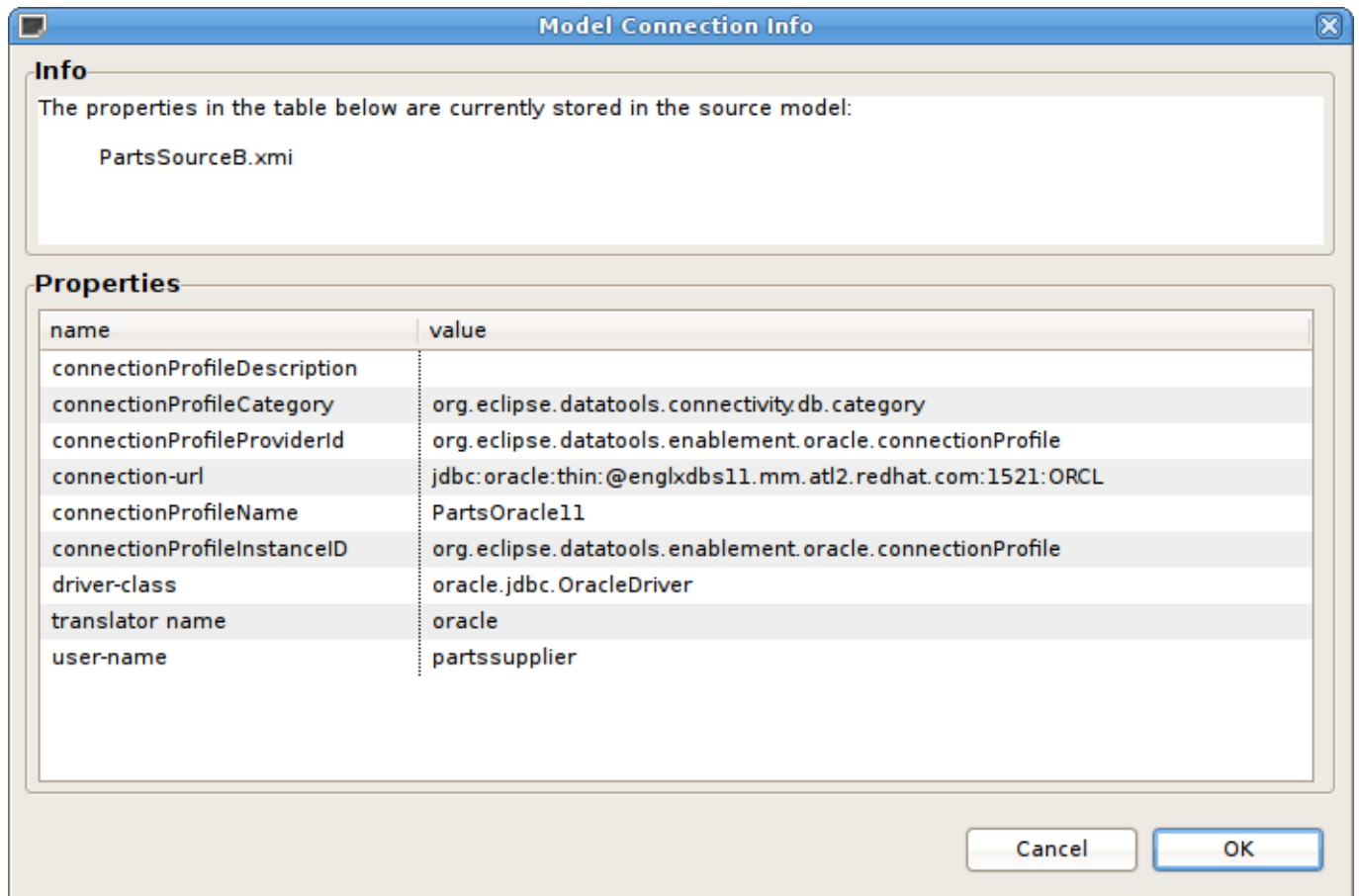


図15.1 Connection Profile Information ダイアログ



### 注記

ソースモデルに関連する接続プロファイルがない場合は、以下のダイアログが表示されます。

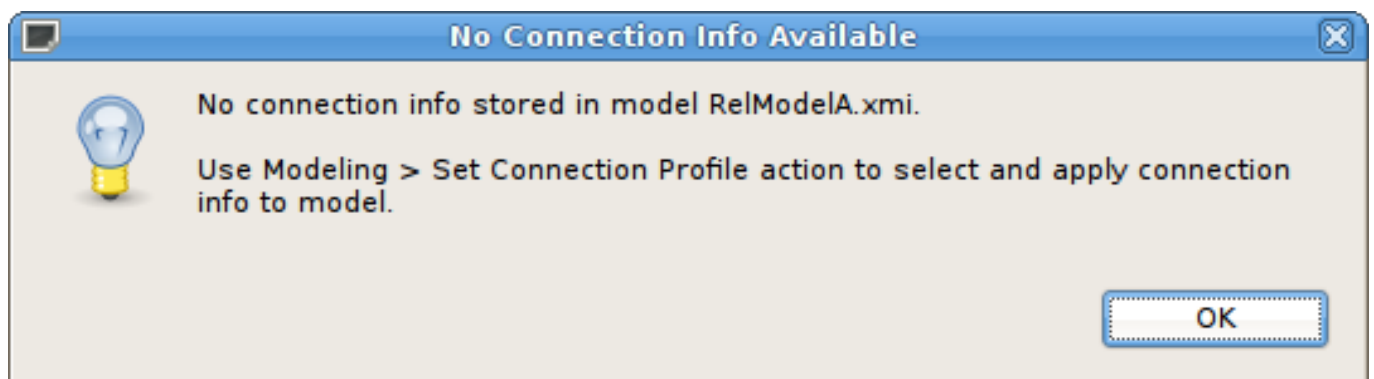


図15.2 No Connection Info ダイアログ


#### 15.1.4. ソースモデルの接続プロファイルの削除

ユーザーとして接続情報 (URL、ユーザー名など) を VDB を介して共有したくない場合があります。Teiid Designer では、**Modeling > Remove Connection Info** と選択してこの接続情報を削除する方法があります。接続情報なしでソースモデルを追加する場合は、ユーザーが適切なトランスレータタイプを提供または選択する必要があります。

## 15.2. モデルのデータプレビュー

### 15.2.1. モデルのデータプレビュー

データの設計や作業を行うとき、データが見れると作業が大変楽になる場合がよくあります。Teiid Designer のプレビューデータ機能はこれを可能にし、物理テーブルや仮想ビューに関わらず、オブジェクトによって記述された情報のプレビューを即座に表示することができます。テーブル、ビュー、手順、または XML ドキュメントを選択するだけで、実際のデータでビューをテストできます。情報をプレビューすると迅速かつ簡単にデータをサンプリングできます。アプリケーションが使用するようなより複雑なクエリーを実行するには、VTP より VDB を実行し、クエリーまたは SQL ステートメントを入力します。

After creating your models, you can test them by using the **Preview Data** action . By selecting a desired table object and executing the action, the results of a simple query will be displayed in the **Preview Results** view. This action is accessible throughout the **Teiid Designer** in various view toolbars and context menus.

データをプレビューする要件は 2 つあります。

1. 選択したオブジェクトは、プレビュー可能なモデルオブジェクトタイプの 1 つである必要があります。
2. モデル依存関係ツリー内のすべてのソースモデルは接続プロファイルを参照する必要があります。

プレビュー可能なモデルオブジェクトには、リレーショナルテーブルおよびビュー (アクセスパターンが関係するテーブルを含む)、リレーショナルプロシージャ、Web サービス演算、XML ドキュメントステージングテーブルが含まれます。

#### 注記

すべての物理ソースモデルが十分な接続情報を参照する限り、仮想テーブル、ビュー、またはプロシージャはプレビュー可能です。

**Preview Data** アクションをクリックした後、**Teiid Designer** ではすべてのソースモデルが接続プロファイルに関連付けられ、必要なパスワードがすべて設定されるようにします。

プレビュー用に選択されたモデルがソースモデルで、そのモデルに関する接続情報が十分でない場合、以下のダイアログが表示され、アクションが停止されます。

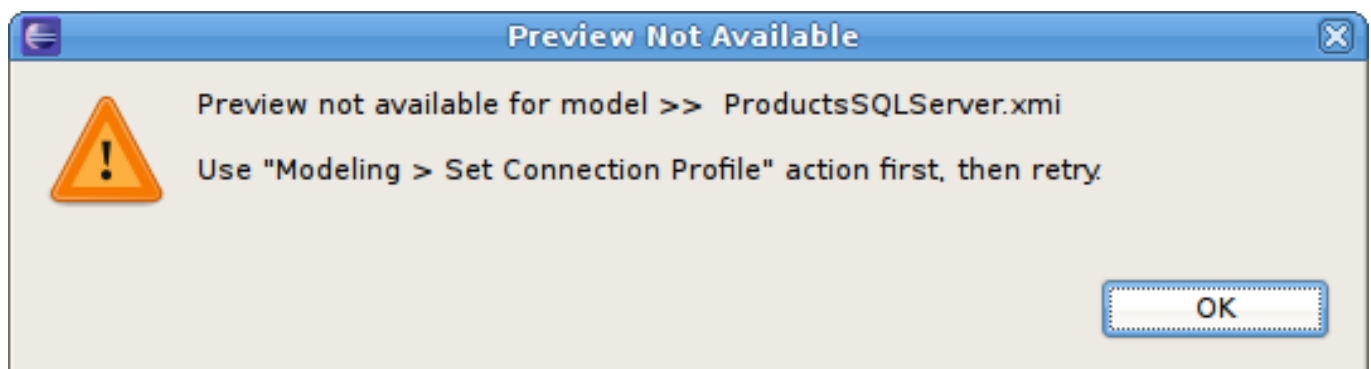


図15.3 Preview Not Available

既存の接続プロファイルから取得できないパスワードを必要とするソースモデルが対応するプロジェクトにある場合、不足しているパスワードごとにユーザーがクエリーされます。



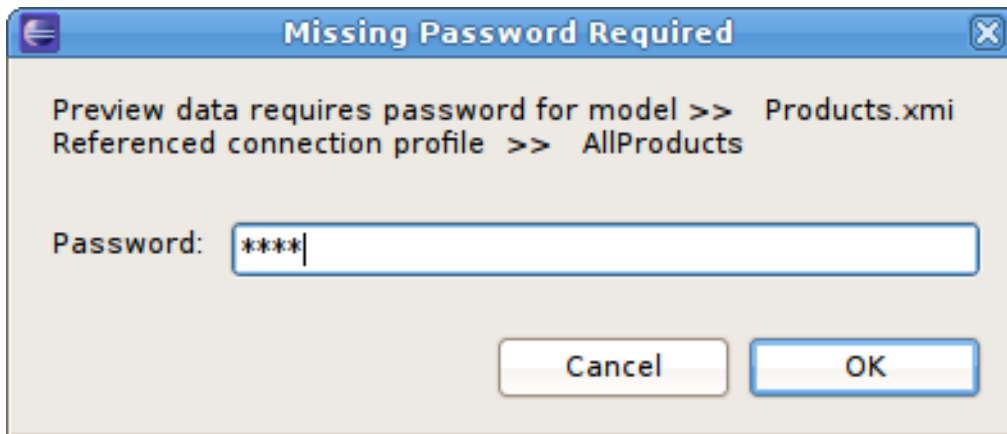




図15.4 Missing Password Required

### トランスフォーメーションのテスト

When editing transformation SQL in the Transformation Editor, a special **SQL Results** data action is provided in the editor toolbar . You can change your transformation SQL, re-validate and preview your the data for your modified SQL. The following sections provide steps for previewing your data. Note that all steps assume that all source models referenced by your models, either directly or through dependencies, are bound to connector bindings.


#### 15.2.2. リレーショナルテーブルまたはビューのプレビュー

リレーショナルテーブル、リレーショナルビュー、またはステージングテーブルをプレビューするには、以下を行います。

1. リレーショナルテーブル、Model Explorer ビューのビュー、または図を選択します。テーブルまたはビューは、ビューモデルやソースモデルにあります。ステージングテーブルは Model Explorer ビューでは表示されないため、マッピング図を開いて選択する必要があります。
2. Right-click select the **Preview Data** action . You can also select the same action in the toolbar of either the Model Explorer view or diagram.
3. クエリーの結果は **Preview Results** ビューに表示されます。ビューは自動的に開かれ、パースペクティブで表示されない場合はフォーカスされます。

#### 15.2.3. アクセスパターンが含まれるリレーショナルテーブルのプレビュー

アクセスパターンが含まれるリレーショナルテーブルまたはビューをプレビューするには、以下を行います。

1. アクセスパターンが含まれるリレーショナルテーブル、Model Explorer ビューのビュー、または図を選択します。テーブルまたはビューは、ビューモデルやソースモデルにあります。
2. Right-click select the **Preview Data** action . You can also select the same action in the toolbar of either the Model Explorer view or diagram.
3. 列入力のダイアログが表示されます。各アクセスパターンを選択し、必要な列の値を入力します。



## 注記

入力したデータが列データタイプ (string、integer など) と一致しない場合は、ダイアログのヘッダーにエラーメッセージが表示されます。

必要な値がすべて入力されたら、**OK** ボタンをクリックしてクエリーを実行します。

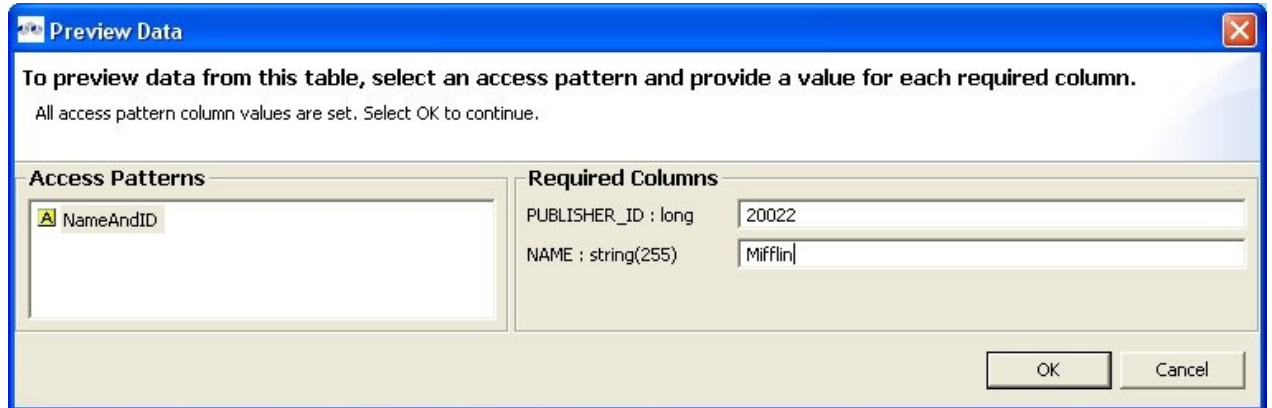



図15.5 アクセスパターンの列入力ダイアログ

- クエリーの結果は **Preview Results** ビューに表示されます。ビューは自動的に開かれ、パースペクティブで表示されない場合はフォーカスされます。

### 15.2.4. リレーショナルプロシージャのプレビュー

リレーショナルプロシージャをプレビューするには、以下を行います。

- Model Explorer ビューのリレーショナルプロシージャまたは図を選択します。プロシージャは、ビューモデルやソースモデルにあります。
- Right-click select the **Preview Data** action . You can also select the same action in the toolbar of either the Model Explorer view or diagram.
- 入力パラメーターの入力ダイアログが表示されます。各パラメーターに有効な値を入力します。



## 注記

入力したデータがパラメーターデータタイプ (string、integer など) と一致しない場合は、ダイアログのヘッダーにエラーメッセージが表示されます。

必要な値がすべて入力されたら、**OK** ボタンをクリックしてクエリーを実行します。

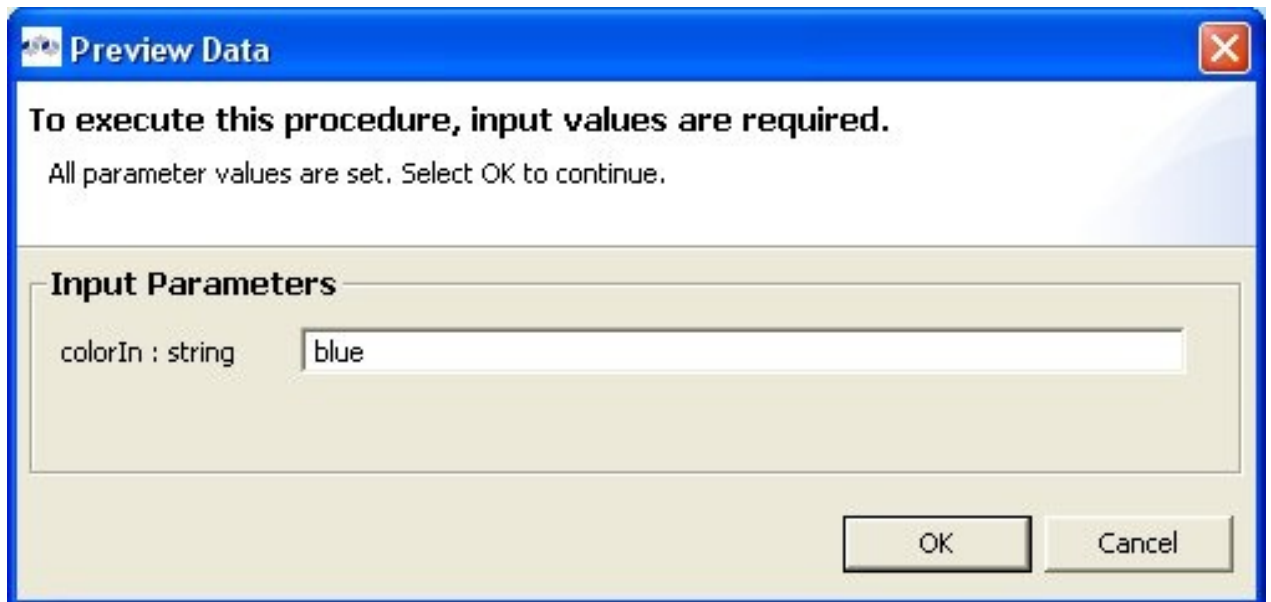



図15.6 プロシージャパラメーター入力のダイアログ

- クエリーの結果は **Preview Results** ビューに表示されます。ビューは自動的に開かれ、パースペクティブで表示されない場合はフォーカスされます。

### 15.2.5. Web サービス演算のプレビュー

Web サービス演算をプレビューするには、以下を行います。

- Model Explorer ビューの Web サービス演算または図を選択します。演算は、ビューモデルやソースモデルにあります。
- Right-click select the **Preview Data** action . You can also select the same action in the toolbar of either the Model Explorer view or diagram.
- 入力パラメーターの入力ダイアログが表示されます。各パラメーターに有効な値を入力します。



#### 注記

入力したデータがパラメーターデータタイプ (string、integer など) と一致しない場合は、ダイアログのヘッダーにエラーメッセージが表示されます。

必要な値がすべて入力されたら、**OK** ボタンをクリックしてクエリーを実行します。

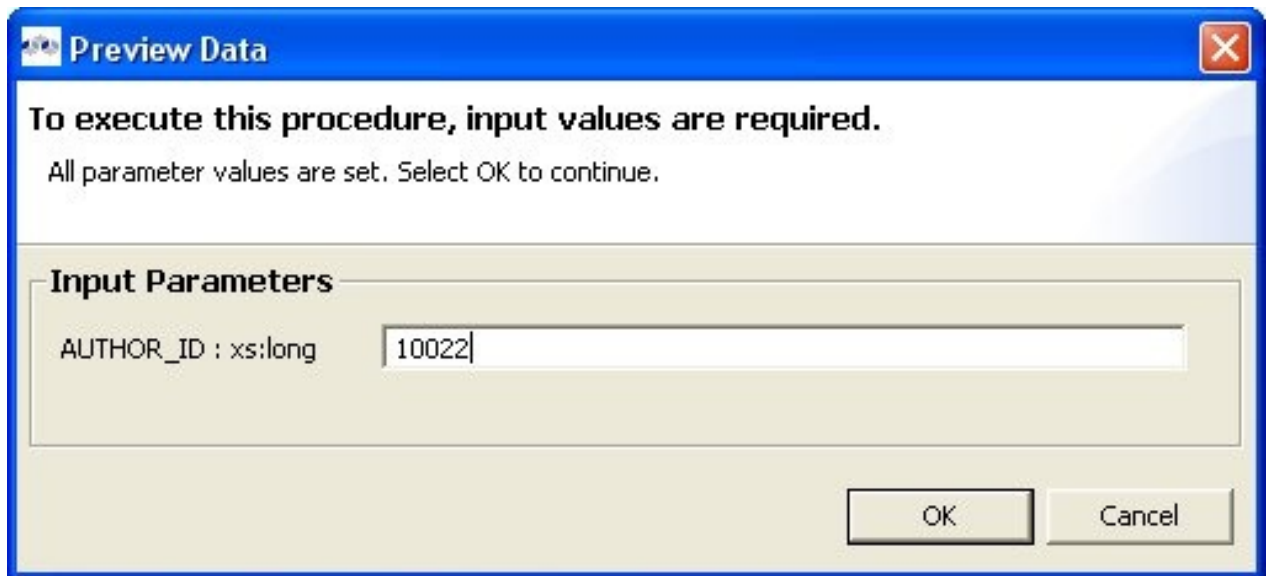


図15.7 プロシージャパラメーター入力のダイアログ

- クエリーの結果は **Preview Results** ビューに表示されます。ビューは自動的に開かれ、パースペクティブで表示されない場合はフォーカスされます。

### 15.2.6. Preview Data の SQL 結果の例

Preview Data の結果は、以下のように Eclipse Datatools SQL Results ビューに表示されます。

**注記**

ツールバーボタンやドロップダウンメニューでは、表示の設定やフィルターオプションを利用できません。

|    | INSTR_ID | NAME                            | TYPE   | ISSUER | EXCHANGE |
|----|----------|---------------------------------|--------|--------|----------|
| 1  | PRD01088 | Novell Incorporated             | Stock  | 1      | NasdaqNM |
| 2  | PRD01089 | Amazon.com, Incorporated        | Stock  | 2      | NasdaqNM |
| 3  | PRD01090 | Juniper Networks, Incorporated  | Stock  | 3      | NasdaqNM |
| 4  | PRD01091 | Red Hat, Incorporated           | Stock  | 4      | NasdaqNM |
| 5  | PRD01092 | Boston Scientific Corporation   | Stock  | 5      | NYSE     |
| 6  | PRD01093 | Inex Pharmaceuticals, Incorp    | Stock  | 1      | Toronto  |
| 7  | PRD01094 | Pfizer, Inc.                    | Stock  | 2      | NasdaqNM |
| 8  | PRD01095 | Cytovax Biotechnologies Incorp  | Stock  | 3      | Toronto  |
| 9  | PRD01096 | Commonwealth Biotechnologies    | Stock  | 4      | NasdaqSC |
| 10 | PRD01097 | British Biotechnology plc       | Stock  | 5      | NasdaqNM |
| 11 | PRD01220 | Unisys Corporation              | Stock  | NULL   | NYSE     |
| 12 | PRD01099 | Honeywell International         | Stock  | 2      | NYSE     |
| 13 | PRD01100 | Hilton Hotels Corporation       | Stock  | 3      | NYSE     |
| 14 | PRD01101 | Hilton Hotels Corporation       | Corpor | 4      | NYSE     |
| 15 | PRD01102 | Mercury Interactive Corporation | Stock  | 5      | NasdaqNM |
| 16 | PRD01103 | Fidelity Freedom Income Fund    | Mutual | 1      | NasdaqSC |
| 17 | PRD01104 | Fidelity Freedom 2000 Fund      | Mutual | 2      | NasdaqSC |

図15.8 SQL Results ビュー

### 15.2.7. 実行計画

Preview Data が実行されると、Teiid Execution Plan も以下のように表示されます。実行計画を取得するには、プレビュー可能なオブジェクトを右クリックし、コンテキストメニューで **Modeling > Show Execution Plan** と選択します。

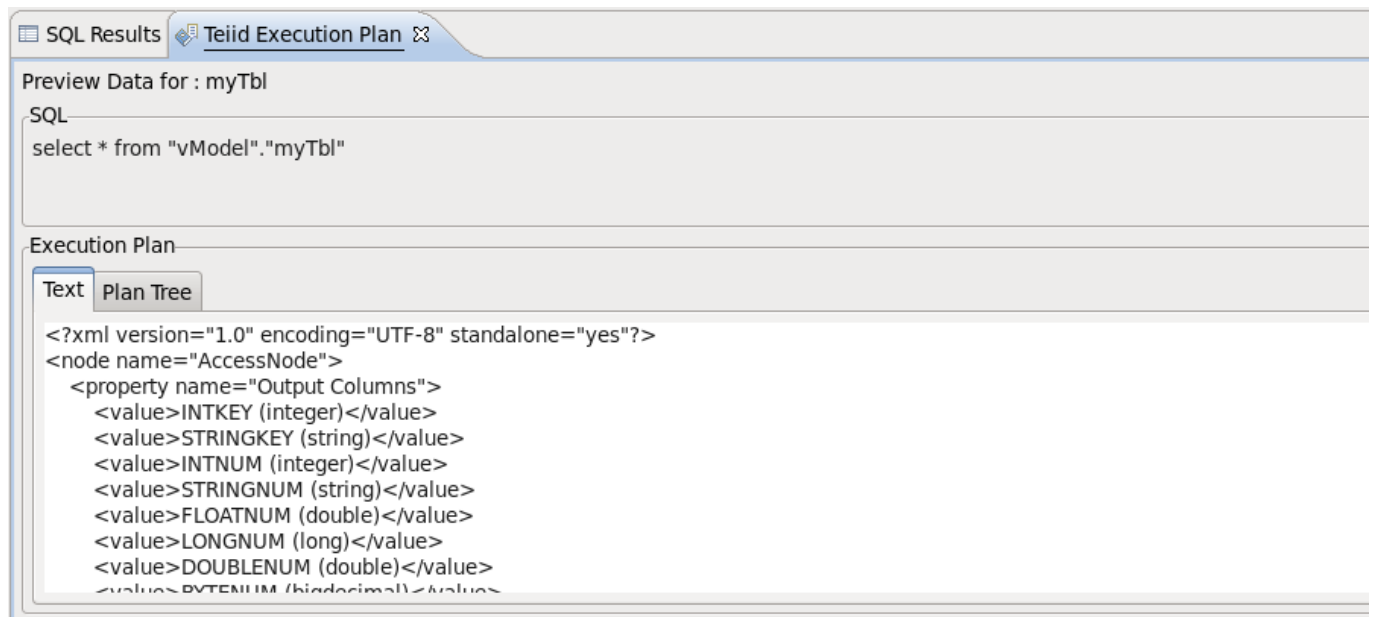


図15.9 Teiid Execution Plan ビュー

## 15.3. VDB を使ったテスト

### 15.3.1. VDB を使ったテスト

Teiid Designer では、VDB を実行して実際のデータをテストおよびクエリーできます。

VDB 実行の要件は次のとおりです。

- ※ デプロイされた有効なデータソースによってサポートされるデプロイされた VDB
- ※ デプロイされた VDB に設定された Teiid Connection Profile のインスタンス。

Teiid Designer では、Deploy VDB および Execute VDB アクションによってこの処理が簡単になります。Deploy VDB を実行すると、選択した VDB が稼働中の Teiid インスタンスにデプロイされます。Execute VDB を実行すると、VDB デプロイメントの実行および Teiid Connection Profile の作成が行われ、Database Development パースペクティブが開かれ、VDB への接続が作成されます。

### 15.3.2. データソースの作成

VDB が実際のデータソースをクエリーできるメカニズムがデータソースです。これらは、データベースまたはソース接続 jar によってサポートされるデプロイされた設定です。VDB 内で参照された各ソースモデルには、デプロイされたデータソースを表す JNDI 名が必要になります。

VDB を作成する際、デプロイされたデータソースは **JBoss Data Virtualization** サーバーには必要ありませんが、VDB をテストする場合はデータソースが存在する必要があります。

Teiid Designer は **Create Data Source** アクションを提供するため、ソースモデルの互換性のあるデータソースを作成することができます。特定のモデルのデータソースを作成するには、ワークスペースでソースモデルを選択し、**Modeling > Create Data Source** と選択します。これにより、ソースモデルから接続プロファイルデータが抽出され、デフォルトの JBoss Data Virtualization サーバーで対応するデータソースが作成されます。

Server ビューからデータソースを作成することもできます。Server ビューで JBoss Data Virtualization サーバーインスタンスを選択し、右クリックしてから **Create Data Source** アクションを選択します。これで **Create Data Source** ダイアログが開きます。

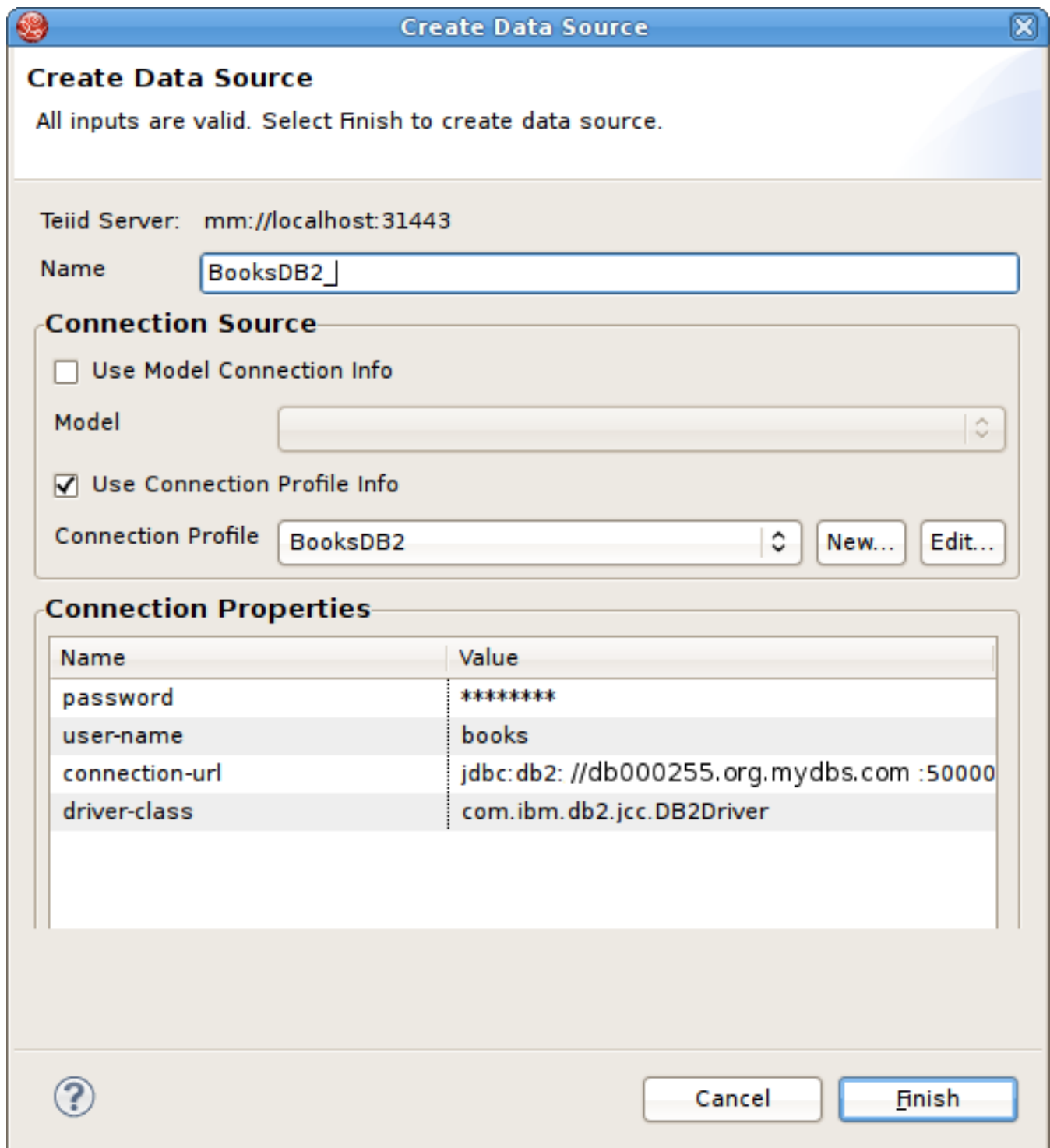


図15.10 Create Data Source ダイアログ

Connection Profile Info オプションのドロップダウンリストから既存の Connection Profile を選択するか、Use Model Info オプションにチェックを入れて connection info を含む既存のソースモデルを選択します。

新しいデータソースを作成したら、対応する JBoss Data Virtualization サーバーの **Data Sources** フォルダに表示されます。

### 15.3.3. Model Explorer からの VDB の実行

JBoss Data Virtualization インスタンスを定義し、サーバーに接続したら以下を行うことができます。

1. Model Explorer の VDB を右クリックし、**Modeling > Execute VDB** と選択します。これにより、選択した VDB が JBoss Data Virtualization にデプロイされ、その VDB に固有の Teiid Connection Profile が作成されます。さらに、Database Development パースペクティブが開かれ、VDB への接続が作成されます。

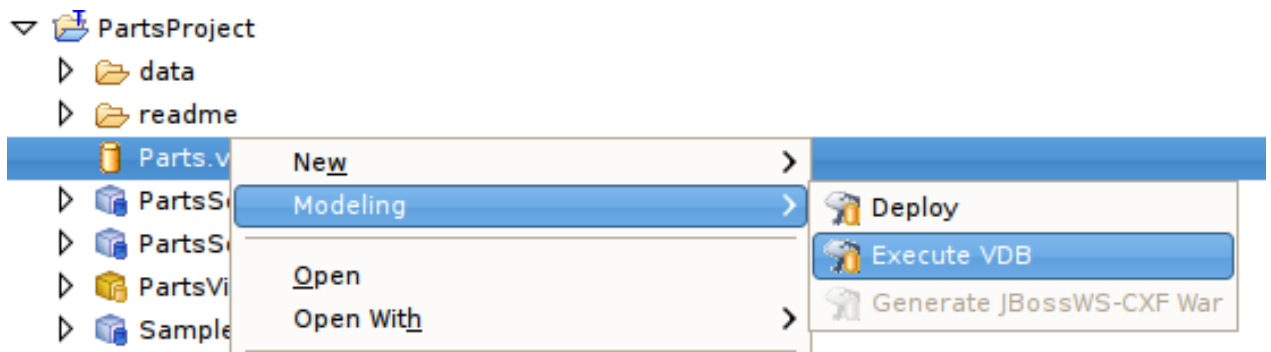


図15.11 Execute VDB

2. 新しい Teiid 接続プロファイルを選択し、右クリックして **Open SQL Scrapbook** を選択し、デザイナー SQL (例: **SELECT \* FROM TableXXXX**) を入力します。すべてのテキストを選択し、右クリックして **Execute Selected Text** を選択します。

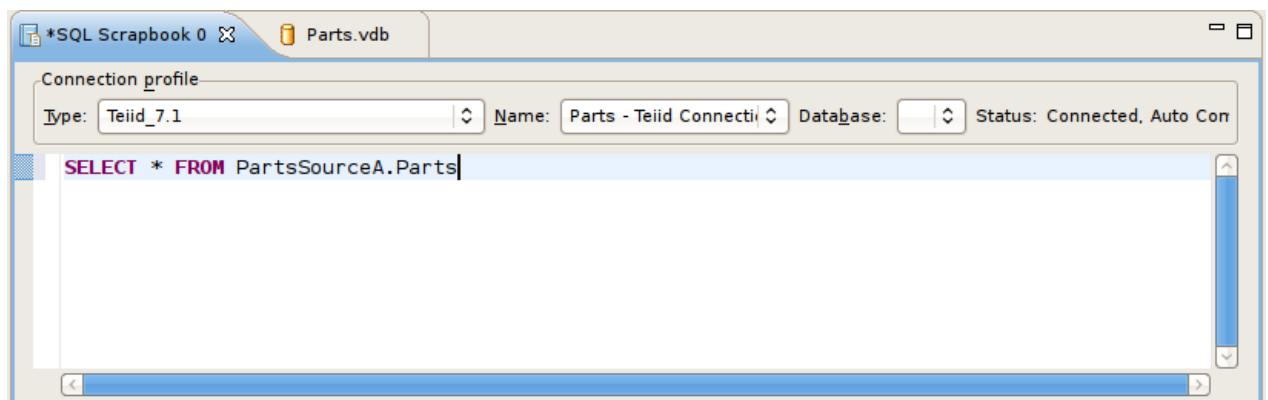


図15.12 SQL Scrapbook エディター

3. クエリーの結果は、**Result1** タブの SQL Results ビューに表示されるはずですが。

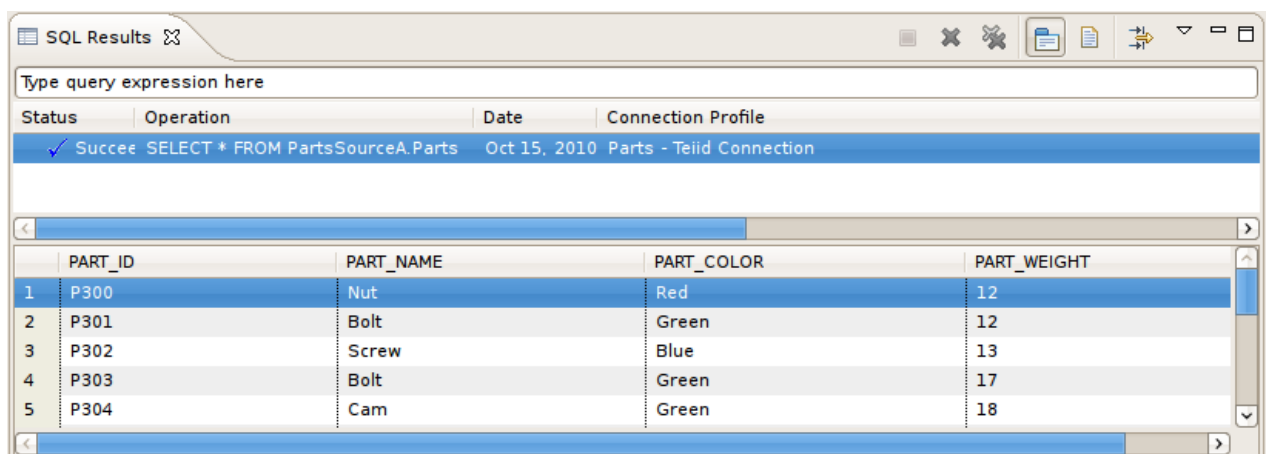


図15.13 SQL Results ビュー

### 15.3.4. Model Explorer からの VDB のデプロイ

最初に VDB をデプロイする場合、Model Explorer で VDB を選択し、Server ビューで接続している JBoss Data Virtualization インスタンスにドラッグアンドドロップすることもできます。また、右クリックして **Modeling > Deploy** と選択することもできます。

デプロイされたら、Server ビューの VDB を選択し、右クリックして **Modeling > Deploy** を選択できます。これにより、その VDB に固有の Teiid Connection Profile が作成され、Database Development パースペクティブが開かれ、VDB への接続が作成されます。ステップ 2 と 3 を繰り返します。



#### 注記

JBoss Data Virtualization インスタンスが定義されていない場合や、デフォルトの JBoss Data Virtualization インスタンスが接続されていない場合、**Modeling > Deploy** と選択すると以下のダイアログが表示されます。

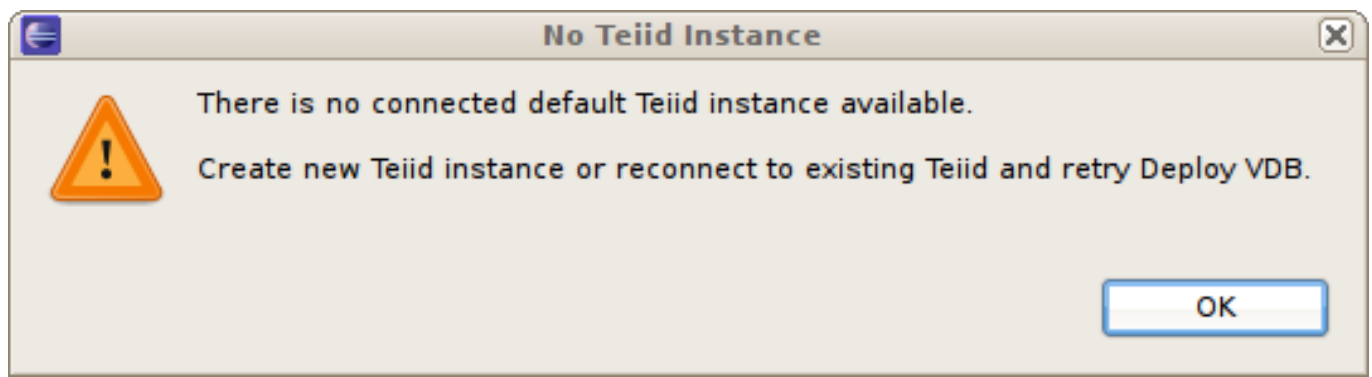


図15.14 No Teiid Instance

### 15.3.5. デプロイされた VDB の実行

手動でデプロイされた VDB を実行するには、以下の手順に従います。

1. **Database Development** パースペクティブを開きます。
2. **Database Connections** フォルダーを選択し、**New** を選択して **New Connection Profile** ダイアログを表示します。





図15.15 New Connection Profile ダイアログ

3. プロファイルの一意な名前を入力し、既存の接続プロファイルタイプを選択した後、**Next** をクリックします。
4. **Teiid Profile Wizard** ページで **New Driver Definition** ボタンを選択し、ファイルシステムの Teiid クライアント jar を見つけて選択します。VDB Name、Host、Port、Username (デフォルトは admin)、および Password (デフォルトは teiid) を使用して URL を設定します。

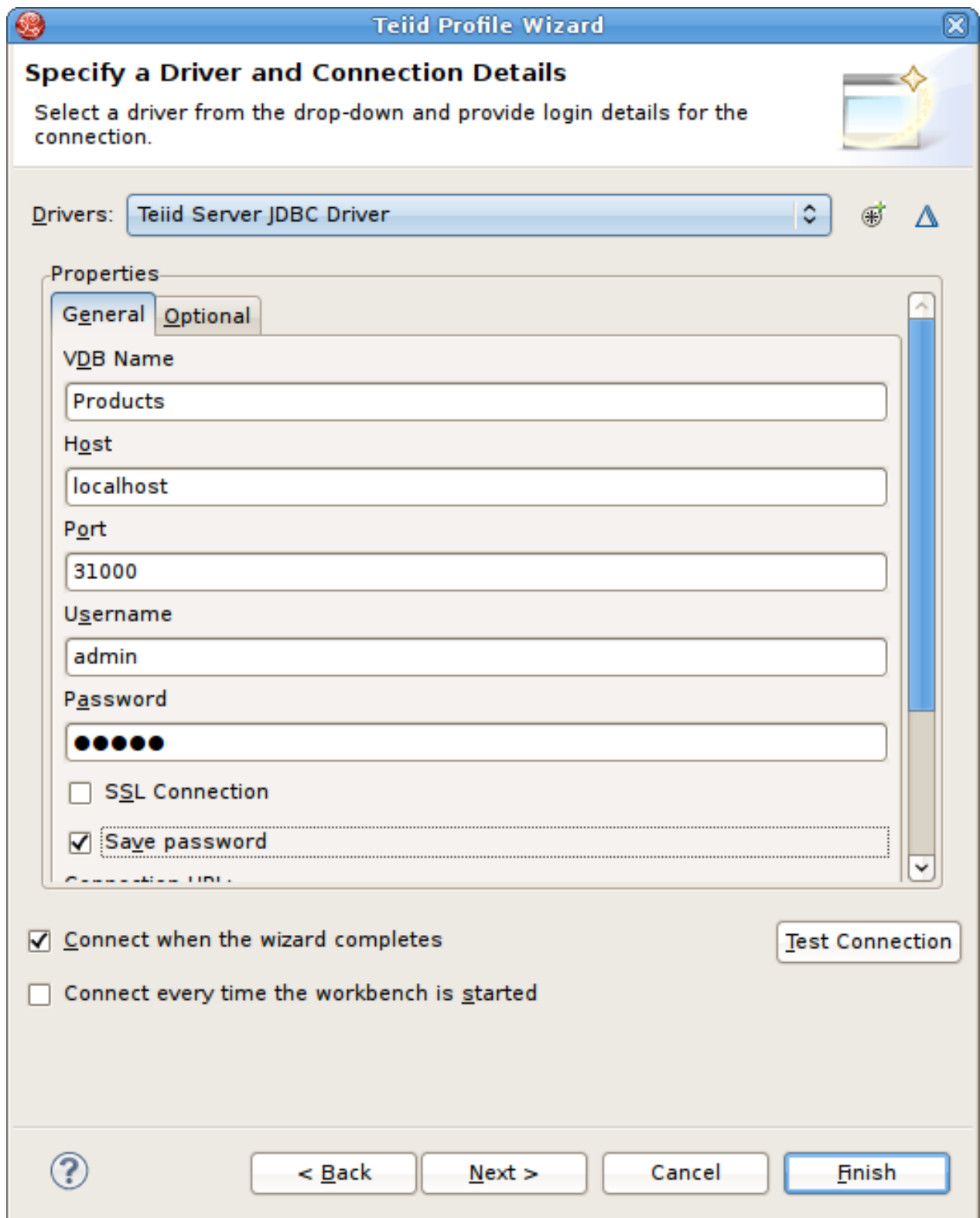


図15.16 Teiid 接続プロファイルのダイアログ

5. **Next** を選択し、新しい Teiid 接続プロファイルの概要を表示します。

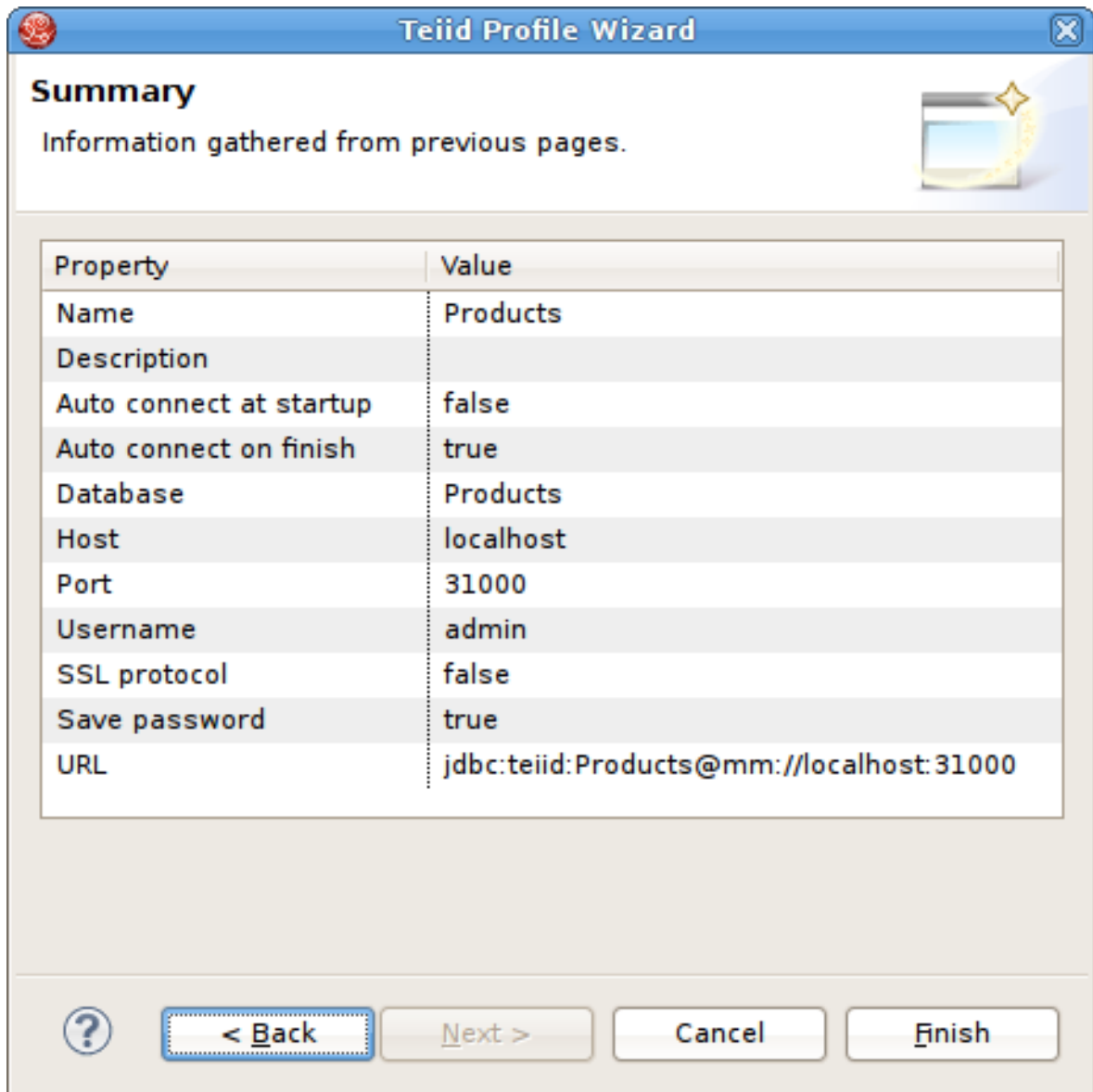


図15.17 Teiid 接続プロファイルの概要

6. **Finish** を選択します。
7. 新しい Teiid 接続プロファイルを選択し、右クリックして **Open SQL Scrapbook** を選択し、デザイナー SQL (例: **SELECT \* FROM TableXXXX**) を入力します。すべてのテキストを選択し、右クリックして **Execute Selected Text** を選択します。

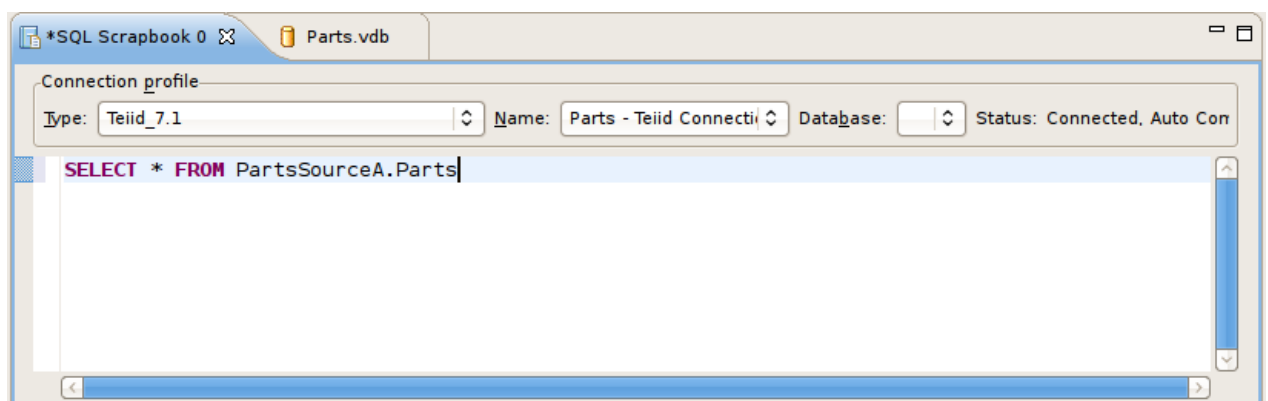


図15.18 SQL Scrapbook エディター

## 第16章 検索

### 16.1. 検索

Eclipses Search メニューの Teiid Designer サブメニューには複数の検索アクションがあります。

Teiid Designer サブメニューの各アクションの詳細は次のとおりです。

- ※ Transformations... - **Transformation Search** ダイアログを起動します。ワークスペースのモデルを検索し、一致する SQL テキストを探すことができます。検索結果はダイアログに表示され、SQL を選択して表示することができます。また、編集のためにトランスフォーメーションを開くこともできます。
- ※ Metadata... - **Search** ダイアログを起動します。ユーザーはワークスペースのモデルを検索し、オブジェクトタイプ、データタイプ、プロパティ値を指定してワークスペースでモデルを検索できます。検索結果は Search Results ビューに表示され、結果をダブルクリックするとそのモデルが適切なエディターで開かれます。
- ※ Find Model Object - オブジェクトの名前のすべてまたは一部を指定して、ワークスペースでオブジェクトを見つけるために使用できる **Find Model Object** ダイアログを起動します。

### 16.2. モデルオブジェクトの検索

Teiid Designer は名前ベースの検索機能を提供し、モデルオブジェクトを迅速に見つけ、表示します。

モデルオブジェクトを検索するには、以下を行います。

1. メインの Teiid Designer ツールバーのアクションを選択するか、メインメニューで **Search > Find Model Object** と選択してアクションを選択します。
2. **Type Object Name** フィールドで言葉または言葉の一部を入力します。ワイルドカード (\*) も使用できます。入力すると、名前と一致するオブジェクトが **Matching Model Objects** リストに表示されます。同じ名前のオブジェクトが複数存在する場合は、オブジェクトの場所またはパスが **Locations** リストに表示されます。
3. 希望の名前を持つオブジェクトが複数存在する場合は、場所の 1 つを選択します。
4. **OK** をクリックします。オブジェクトのモデルに対してエディターが開かれていない場合は、エディターが開かれます。オブジェクトが図に表示され (該当する場合)、選択されます。

### 16.3. トランスフォーメーション SQL の検索

Teiid Designer は、トランスフォーメーション SQL テキストに存在する文字列値を検索する機能を提供します。

トランスフォーメーション SQL で文字列値を検索するには、以下を実行します。

1. Teiid Designer メインメニューで **Search > Transformations...** とクリックし、**Search Transformations** ダイアログを開きます。
2. **Find:** フィールドで文字列のセグメントを指定し、大文字と小文字の区別を指定または変更します。
3. **Perform Search** ボタンを選択します。文字列が含まれる SQL テキストが含まれるトランスフォーメーションオブジェクトが結果セクションに表示されます。

個別のオブジェクトを選択し、SQL を表示できます。テーブルまたはビューが更新をサポートし、insert、update、または delete SQL が存在する場合、オブジェクトを展開し、以下のように個別の SQL タイプを選択することができます。

選択したオブジェクトとその SQL を **Model Editor** で表示するには **Edit** ボタンをクリックします。エディターが開かれていない場合は開かれます。エディターが開かれている場合はそのタブが選択されます。さらに、**Transformation Editor** が開かれ、Find/Replace (**Ctrl-F**) を実行して検索対象のテキスト文字列をハイライト表示し、必要な場合は SQL を編集できます。

## 16.4. メタデータプロパティによるモデルの検索

**Teiid Designer** は、1 つ以上のメタデータプロパティ値によって分類されるモデルオブジェクトを見つける検索機能を提供します。

メタデータを使用してモデルを検索するには、以下を実行します。

1. メインの Teiid Design ツールバー **Search > Metadata...** と選択し、**Search** ダイアログを開きます。
2. Object Type、Data Type、および Properties の検索オプションを指定します。
3. **Search** をクリックします。検索が実行され、結果が **Search Results View** に表示されます。ビューが開かれていない場合は自動的に開かれます。

## 付録A サポートされる構成

### A.1. サポートされるデータソースおよびトランスレーター

以下の表は Red Hat によってサポートされるデータソースおよびトランスレーターの一覧になります。

表A.1 サポートされるデータソースおよびトランスレーター

| データソース                                  | トランスレーター             | サポートされる DVバージョン    | ドライバー                              |
|---|----------------------|--------------------|------------------------------------|
| Apache Hive 12                          | -                    | 6.0+               | -                                  |
| Apache Solr                             | solr                 | 6.1+               | -                                  |
| Cloudera Hadoop                         | -                    | 6.1+               | -                                  |
| EDS 5.x                                 | teiid                | 6.0+               | -                                  |
| ファイル – 区切り、固定長                          | file                 | 6.0+               | -                                  |
| 汎用データソース -JDBC ansi                     | jdbc-ansi            | 6.0+               | -                                  |
| 汎用データソース -JDBC simple                   | jdbc-simple          | 6.0+               | -                                  |
| Google Spreadsheet                      | -                    | 6.0+               | -                                  |
| Greenplum 4.x                           | postgresql           | 6.0+               | -                                  |
| Hortonworks Hadoop                      | -                    | 6.1.+              | -                                  |
| HSQL (テスト/例のみ)                          | -                    | -                  | -                                  |
| IBM DB2 10                              | db2                  | 6.1+               | Universal Driver v4.x              |
| IBM DB2 9.7                             | db2                  | 6.0+               | Universal Driver v4.x              |
| Ingres 10                               | ingres               | 6.0+               | -                                  |
| Intel Hadoop                            | -                    | 6.1+               | -                                  |
| JBoss Data Grid 6.4 (リモートクライアント-hotrod) | infinispan-cache-dsl | 6.2+               | -                                  |
| JBoss Data Grid 6.4 (ライブラリーモード)         | infinispan-cache     | 6.0 - post GA、6.1+ | -                                  |
| LDAP/ActiveDirectory v3                 | ldap                 | 6.0+               | -                                  |
| Mainframe (CICS,IMS,VSAM)               | -                    | 6.0+               | -                                  |
| MariaDB                                 | mysql5               | 6.1+               | -                                  |
| ModeShape/JCR 3.1                       | -                    | 6.0+               | -                                  |
| MongoDB 2.2                             | mongodb              | 6.0                | post GA、6.1+ -                     |
| MS Access 2010                          | -                    | 6.0+               | -                                  |
| MS Access 2013                          | -                    | 6.0+               | -                                  |
| MS Excel 2010                           | excel                | 6.0+               | -                                  |
| MS Excel 2013                           | excel                | 6.0+               | -                                  |
| MS SQL Server 2008                      | sqlserver            | 6.0+               | Microsoft SQL Server JDBC Driver 4 |
| MS SQL Server 2012                      | sqlserver            | 6.0+               | Microsoft SQL Server JDBC Driver 4 |
| MySQL 5.1                               | mysql5               | 6.0+               | V5.1                               |
| MySQL 5.5                               | mysql5               | 6.0+               | V5.5                               |
| Netezza 6.0.2                           | netezza              | 6.0+               | -                                  |
| Oracle 10g R2                           | oracle               | 6.0+               | Oracle JDBC Driver v10             |

| データソース                              | トランスレーター       | サポートされるDVバージョン     | ドライバー                  |
|-------------------------------------|----------------|--------------------|------------------------|
| Oracle 11g RAC                      | oracle         | 6.0+               | Oracle JDBC Driver v11 |
| Oracle 12c                          | oracle         | 6.0 - post GA、6.1+ | Oracle JDBC Driver v12 |
| PostgreSQL 8.4                      | postgresql     | 6.0+               | -                      |
| PostgreSQL 9.2                      | postgresql     | 6.0+               | -                      |
| HTTP 上の REST/JSON                   | ws             | -                  | -                      |
| RHEL 5.5/6 PostgreSQL config        | -              | 6.0+               | -                      |
| Salesforce.com API 22               | salesforce     | 6.0+               | -                      |
| SAP Netweaver Gateway (OData)       | sap-nw-gateway | 6.1+               | -                      |
| SAP Service Registry をデータソースとしてサポート | -              | 6.2+               | -                      |
| Sybase ASE 15                       | sybase         | 6.0+               | jConnect JDBC3.0 v7    |
| Teradata Express 12                 | teradata       | 6.0+               | -                      |
| Webservices                         | ws             | 6.0+               | -                      |
| XML ファイル                            | FILE           | 6.0+               | -                      |



### 注記

MS Excel は書き込みプロシージャがあるためサポートされます。

## A.2. JBoss Data Virtualization ランタイムでの Teiid Designer メタデータ使用要件

Teiid Designer によって公開されるメタデータを基に、以下の表は必要なフィールドと、JBoss Data Virtualization ランタイムでフィールドの情報がどのように使用されるかを表しています。

表A.2 テーブルのデータ使用

| テーブル         | Type | Teiid Designer | Metadata API | 必要性 | 説明   |
|--------------|------|----------------|--------------|-----|--|
| FullName     | 文字列  | はい             | はい           | はい  | テーブルの名前  |
| NameInSource | 文字列  | はい             | はい           | はい  | ソースシステムでのテーブルの名前。ビューでは空にでき、さまざまなユースケースに使用されます。 |
| Cardinality  | 整数   | はい             | はい           | はい  | ソースノードアクセスのコストを算出するために使用されます。                  |

| テーブル                | Type   | Teiid Designer | Metadata API     | 必要性 | 説明  |
|---------------------|--------|----------------|------------------|-----|---|
| TableType           | 整数     | はい             | はい               | はい  | Table,View,Document,XmlMappingClass,XmlStagingTable,MaterializedTable |
| IsVirtual           | ブール値   | はい             | はい               | はい  | ソーステーブルまたはビューであるかを確認するために使用されます。                                      |
| IsSystem            | ブール値   | はい             | はい               | なし  | システムメタデータのみで使用されます。   |
| IsMaterialized      | ブール値   | はい             | はい               | はい  | テーブルがマテリアライズされているかを識別します。   |
| SupportsUpdate      | ブール値   | はい             | はい               | はい  | テーブルの更新を許可します。  |
| PrimaryKeyID        | 文字列    | はい             | KeyRecord        | はい  | 一時テーブルでのインデックスの作成に使用され、デフォルトの update/delete プロシージャを作成します。             |
| ForeignKeyIDs       | コレクション | はい             | List<ForeignKey> | はい  | クエリーの計画に使用されます (ルールの raise access)                                    |
| IndexIDs            | コレクション | はい             | List<KeyRecord>  | はい  | 一時テーブルまたは計画でのインデックスの作成に使用されます。  |
| UniqueKeyIDs        | コレクション | はい             | List<KeyRecord>  | はい  | クエリーの計画に使用されます。   |
| AccessPatterns      | コレクション | はい             | List<KeyRecord>  | はい  | クエリーで基準を強制するために使用されます。  |
| MaterializedTableID | 文字列    | はい             | Table            | はい  | マテリアライズテーブルへの参照   |
| insertEnabled       | ブール値   | **             | はい               | はい  | ビューに対して insert プロシージャが有効になっているかをチェックするフラグ。                            |



| テーブル                  | Type   | Teiid Designer | Metadata API | 必要性 | 説明   |
|-----------------------|--------|----------------|--------------|-----|--|
| deleteEnabled         | ブール値   | **             | はい           | はい  | ビューに対して delete プロシージャが有効になっているかをチェックするフラグ。 |
| updateEnabled         | ブール値   | **             | はい           | はい  | ビューに対して update プロシージャが有効になっているかをチェックするフラグ。 |
| Select Transformation | 文字列    | **             | はい           | はい  | ビューの場合の Select のトランスフォーメーション。              |
| Insert Plan           | 文字列    | **             | はい           | はい  | ビューの場合の Insert のトランスフォーメーション。              |
| Update Plan           | 文字列    | **             | はい           | はい  | ビューの場合の Update のトランスフォーメーション。              |
| Delete Plan           | 文字列    | **             | はい           | はい  | ビューの場合の Delete のトランスフォーメーション。              |
| Bindings              | コレクション | **             | はい           | はい  | XML ドキュメント。                                |
| SchemaPaths           | コレクション | **             | はい           | はい  | XML ドキュメント。                                |

表A.3 列のデータ使用

| 列            | Type | Teiid Designer | Metadata API | 必要性 | 説明                               |
|--------------|------|----------------|--------------|-----|----------------------------------|
| FullName     | 文字列  | はい             | はい           | はい  | 列の名前                             |
| NameInSource | 文字列  | はい             | はい           | はい  | ソースシステムの列の名前。                    |
| IsSelectable | ブール値 | はい             | はい           | はい  | 列は select で許可されます。               |
| IsUpdatable  | ブール値 | はい             | はい           | はい  | 列は Update、Insert、Delete で許可されます。 |
| NullType     | 整数   | はい             | はい           | はい  | null 値が許可される場合は検証で使用されます。        |

| 列                              | Type | Teiid Designer | Metadata API | 必要性 | 説明                                    |
|--------------------------------|------|----------------|--------------|-----|---------------------------------------|
| IsAutoIncrementable            | ブール値 | はい             | はい           | はい  | 値が必要であるかどうかを検証するため insert 処理中に使用されます。 |
| IsCaseSensitive                | ブール値 | はい             | はい           | ??  | ??                                    |
| IsSigned                       | ブール値 | はい             | はい           | ??  | システムメタデータで使用されます。                     |
| IsCurrency                     | ブール値 | はい             | はい           | なし  | システムメタデータのみで使用されます。                   |
| IsFixedLength                  | ブール値 | はい             | はい           | なし  | システムメタデータのみで使用されます。                   |
| IsTransformationInputParameter | ブール値 | はい             | ??           | ??  | ??                                    |
| SearchType                     | 整数   | はい             | はい           | はい  | ソースの機能を定義するために使用されます。                 |
| Length                         | 整数   | はい             | はい           | ??  | システムメタデータで使用されます。                     |
| Scale                          | 整数   | はい             | はい           | ??  | システムメタデータで使用されます。                     |
| Precision                      | 整数   | はい             | はい           | ??  | システムメタデータで使用されます。                     |
| CharOctetLength                | 整数   | はい             | はい           | なし  | システムメタデータのみで使用されます。                   |
| Radix                          | 整数   | はい             | はい           | ??  | システムメタデータで使用されます。                     |
| DistinctValues                 | 整数   | はい             | はい           | はい  | コストの計算に使用されます。システムメタデータ。              |
| NullValues                     | 整数   | はい             | はい           | はい  | コストの計算に使用されます。システムメタデータ。              |
| MinValue                       | 文字列  | はい             | はい           | はい  | コストの計算に使用されます。システムメタデータ。              |
| MaxValue                       | 文字列  | はい             | はい           | はい  | コストの計算に使用されます。システムメタデータ。              |

| 列                 | Type | Teiid Designer | Metadata API | 必要性 | 説明                                     |
|-------------------|------|----------------|--------------|-----|--|
| Format            | 文字列  | はい             | はい           | なし  | システムメタデータのみで使用されます。                    |
| RuntimeType       | 文字列  | はい             | DataType     | はい  | データタイプ                                 |
| NativeType        | 文字列  | はい             | はい           | はい  | トランスレータはこのフィールドを使用してさらに計画できます。         |
| DatatypeObject ID | 文字列  | はい             | ??           | ??  |  |
| DefaultValue      | 文字列  | はい             | はい           | はい  | 値の指定がない場合にInsert およびプロシージャ実行演算に使用されます。 |
| 役職                | 整数   | はい             | はい           | はい  | インデックスの計算に使用されます。                      |

表A.4 プライマリーキーのデータ使用

| プライマリーキー      | Type   | Teiid Designer | Metadata API | 必要性 | 説明                   |
|---------------|--------|----------------|--------------|-----|----------------------|
| FullName      | 文字列    |                |              |     | KeyRecord、Table を参照。 |
| NameInSource  | 文字列    |                |              |     |                      |
| ColumnIDs     | コレクション |                |              |     |                      |
| ForeignKeyIDs | コレクション |                |              |     | KeyRecord を拡張します。    |

表A.5 一意キーのデータ使用

| 一意キー          | Type   | Teiid Designer | Metadata API | 必要性 | 説明                   |
|---------------|--------|----------------|--------------|-----|----------------------|
| FullName      | 文字列    |                |              |     | KeyRecord、Table を参照。 |
| NameInSource  | 文字列    |                |              |     |                      |
| ColumnIDs     | コレクション |                |              |     |                      |
| ForeignKeyIDs | コレクション |                |              |     |                      |

表A.6 インデックスのデータ使用

| インデックス       | Type   | Teiid Designer | Metadata API | 必要性 | 説明                   |
|--------------|--------|----------------|--------------|-----|----------------------|
| FullName     | 文字列    |                |              |     | KeyRecord、Table を参照。 |
| NameInSource | 文字列    |                |              |     |                      |
| ColumnIDs    | コレクション |                |              |     |                      |

表A.7 アクセスパターンのデータ使用

| アクセサパターン     | Type   | Teiid Designer | Metadata API | 必要性 | 説明                   |
|--------------|--------|----------------|--------------|-----|----------------------|
| FullName     | 文字列    |                |              |     | KeyRecord、Table を参照。 |
| NameInSource | 文字列    |                |              |     |                      |
| ColumnIDs    | コレクション |                |              |     |                      |

表A.8 結果セットのデータ使用

| 結果セット        | Type   | Teiid Designer | Metadata API | 必要性 | 説明            |
|--------------|--------|----------------|--------------|-----|---------------|
| FullName     | 文字列    |                |              |     | DataType を参照。 |
| NameInSource | 文字列    |                |              |     |               |
| ColumnIDs    | コレクション |                |              |     |               |

表A.9 外部キーのデータ使用

| 外部キー         | Type   | Teiid Designer | Metadata API | 必要性 | 説明                   |
|--------------|--------|----------------|--------------|-----|----------------------|
| FullName     | 文字列    |                |              |     | KeyRecord、Table を参照。 |
| NameInSource | 文字列    |                |              |     |                      |
| ColumnIDs    | コレクション |                |              |     |                      |
| UniqueKeyID  | 文字列    |                |              |     |                      |

表A.10 データタイプのデータ使用

| データタイプ          | Type | Teiid Designer | Metadata API | 必要性 | 説明                  |
|-----------------|------|----------------|--------------|-----|---------------------|
| FullName        | 文字列  |                |              | なし  | システムメタデータのみに使用されます。 |
| NameInSource    | 文字列  |                |              | なし  | システムメタデータのみに使用されます。 |
| Length          | 整数   |                |              | なし  | システムメタデータのみに使用されます。 |
| PrecisionLength | 整数   |                |              | なし  | システムメタデータのみに使用されます。 |
| Scale           | 整数   |                |              | なし  | システムメタデータのみに使用されます。 |
| Radix           | 整数   |                |              | なし  | システムメタデータのみに使用されます。 |
| IsSigned        | ブール値 |                |              | なし  | システムメタデータのみに使用されます。 |
| IsAutoIncrement | ブール値 |                |              | なし  | システムメタデータのみに使用されます。 |

| データタイプ          | Type   | Teiid Designer | Metadata API | 必要性 | 説明                          |
|-----------------|--------|----------------|--------------|-----|-----------------------------|
| IsCaseSensitive | ブール値   |                |              | なし  | システムメタデータのみに使用されます。         |
| Type            | 整数     |                |              | なし  | システムメタデータのみに使用されます。         |
| SearchType      | 整数     |                |              | なし  | システムメタデータのみに使用されます。         |
| NullType        | 整数     |                |              | なし  | システムメタデータのみに使用されます。         |
| JavaClassName   | 文字列    |                |              | はい  | Java クラス名を基にしてランタイムへマップします。 |
| RuntimeTypeName | 文字列    |                |              | なし  | システムメタデータのみに使用されます。         |
| DatatypeID      | 文字列    |                |              | なし  | システムメタデータのみに使用されます。         |
| BaseTypeID      | 文字列    |                |              | なし  | システムメタデータのみに使用されます。         |
| PrimitiveTypeID | 文字列    |                |              | なし  | システムメタデータのみに使用されます。         |
| VarietyType     | 整数     |                |              | なし  | システムメタデータのみに使用されます。         |
| VarietyProps    | コレクション |                |              | なし  | システムメタデータのみに使用されます。         |

表A.11 プロシージャーのデータ使用

| プロシージャー       | Type   | Teiid Designer | Metadata API | 必要性 | 説明                               |
|---------------|--------|----------------|--------------|-----|----------------------------------|
| FullName      | 文字列    | はい             | はい           | はい  | 列の名前                             |
| NameInSource  | 文字列    | はい             | はい           | はい  | ソースシステムの列の名前。                    |
| IsFunction    | ブール値   |                | はい           | はい  | 関数であるかを決定します。                    |
| IsVirtual     | ブール値   |                | はい           | はい  | 関数の場合、ユーザー定義関数 else ストアドプロシージャー。 |
| ParametersIDs | コレクション |                | はい           | はい  | パラメーターリスト                        |
| ResultSetID   | 文字列    |                | はい           | はい  | 結果セットの列                          |

| プロシージャー     | Type | Teiid Designer | Metadata API | 必要性 | 説明                                     |
|-------------|------|----------------|--------------|-----|--|
| UpdateCount | 整数   |                | はい           | はい  | 更新数は更新されたソースの数を定義します。仮想プロシージャーのみ該当します。 |

表A.12 プロシージャーパラメーターのデータ使用

| プロシージャー<br>パラメーター    | Type | Teiid Designer | Metadata API | 必要性 | 説明   |
|----------------------|------|----------------|--------------|-----|--|
| ObjectID             | 文字列  |                |              |     | 列と同じ。                                      |
| FullName             | 文字列  |                |              |     | 列と同じ。                                      |
| nameInSource         | 文字列  |                |              |     | 列と同じ。                                      |
| defaultValue         | 文字列  |                |              |     | 列と同じ。                                      |
| RuntimeType          | 文字列  |                |              |     | 列と同じ。                                      |
| DatatypeObject<br>ID | 文字列  |                |              |     | 列と同じ。                                      |
| Length               | 整数   |                |              |     | 列と同じ。                                      |
| Radix                | 整数   |                |              |     | 列と同じ。                                      |
| Scale                | 整数   |                |              |     | 列と同じ。                                      |
| NullType             | 整数   |                |              |     | 列と同じ。                                      |
| Precision            | 整数   |                |              |     | 列と同じ。                                      |
| 役職                   | 整数   |                |              |     | 列と同じ。                                      |
| Type                 | 文字列  |                |              | はい  | パラメーターがIN、OUT、RETURNであることを定義します。           |
| オプション                | ブール値 |                |              | なし  | パラメーターが任意であるかどうかを定義します。システムメタデータのみで使用されます。 |

表A.13 SQL トランスフォーメーションのデータ使用

| SQL<br>TRANSFORM<br>ATION(**) | Type | Teiid Designer | Metadata API | 必要性 | 説明                        |
|-------------------------------|------|----------------|--------------|-----|---------------------------|
| VirtualGroupName              | 文字列  | はい             | なし           | はい  | テーブル、テーブルで定義されたプロパティーを参照。 |
| TransformedObjectID           | 文字列  | はい             | なし           | はい  | テーブル、テーブルで定義されたプロパティーを参照。 |

| SQL TRANSFORMATION(**)  | Type   | Teiid Designer | Metadata API | 必要性 | 説明                        |
|-------------------------|--------|----------------|--------------|-----|---------------------------|
| Transformation ObjectID | 文字列    | はい             | なし           | はい  | テーブル、テーブルで定義されたプロパティーを参照。 |
| Transformation Sql      | 文字列    | はい             | なし           | はい  | テーブル、テーブルで定義されたプロパティーを参照。 |
| Bindings                | コレクション | はい             | なし           | はい  | テーブル、テーブルで定義されたプロパティーを参照。 |
| SchemaPaths             | コレクション | はい             | なし           | はい  | テーブル、テーブルで定義されたプロパティーを参照。 |

表A.14 VDB のデータ使用

| VDB              | Type   | Teiid Designer | Metadata API | 必要性 | 説明                   |
|------------------|--------|----------------|--------------|-----|----------------------|
| FullName         | 文字列    | はい             | vdb.xml      | はい  | VDB の名前              |
| NameInSource     | 文字列    | ??             | なし           | なし  | 不要                   |
| バージョン            | 文字列    | はい             | vdb.xml      | はい  | VDB バージョン            |
| Identifier       | 文字列    | はい             | なし           | なし  | 不要                   |
| 説明               | 文字列    | はい             | vdb.xml      | なし  | システムメタデータによって使用されます。 |
| ProducerName     | 文字列    | はい             | なし           | なし  | 不要                   |
| ProducerVersion  | 文字列    | はい             | なし           | なし  | 不要                   |
| プロバイダー           | 文字列    | はい             | なし           | なし  | 不要                   |
| TimeLastChanged  | 文字列    | はい             | なし           | なし  | 不要                   |
| TimeLastProduced | 文字列    | はい             | なし           | なし  | 不要                   |
| ModelIDs         | コレクション | はい             | vdb.xml      | はい  | VDB のモデルリストを定義します。   |

表A.15 アノテーションのデータ使用

| アノテーション      | Type | Teiid Designer | Metadata API | 必要性 | 説明                         |
|--------------|------|----------------|--------------|-----|----------------------------|
| FullName     | 文字列  | はい             | はい           | なし  | システムメタデータ、プロシージャパラメーターの説明。 |
| NameInSource | 文字列  | はい             | なし           | なし  | 不要                         |

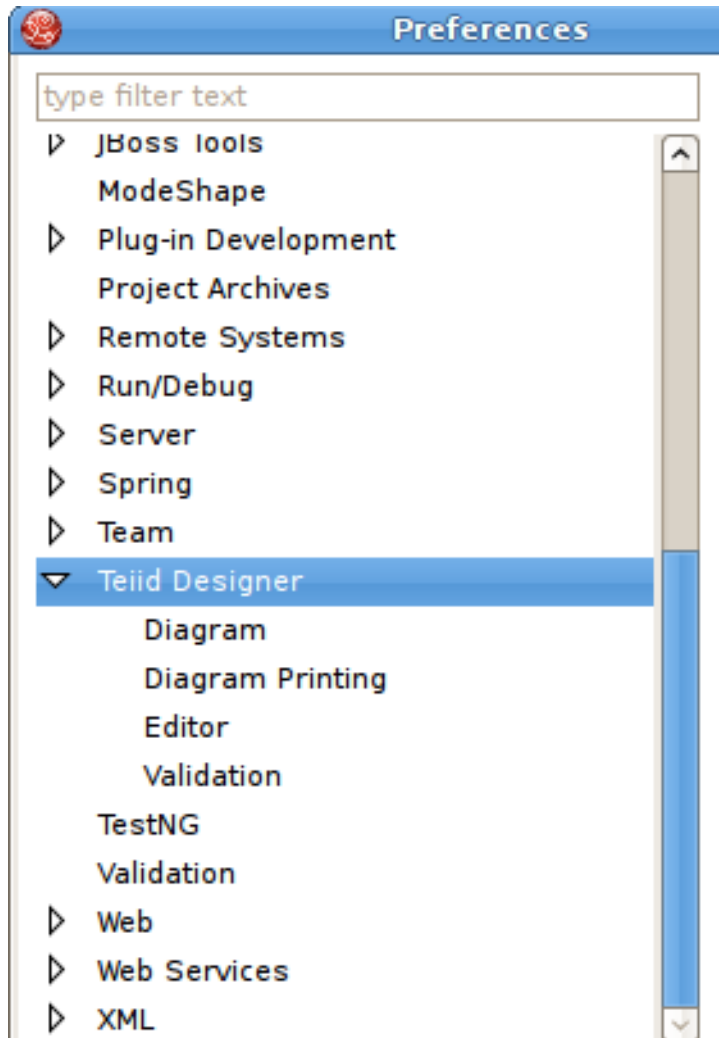
| アノテーション | Type | Teiid Designer | Metadata API | 必要性 | 説明 |
|---------|------|----------------|--------------|-----|----|
| 説明      | 文字列  | はい             | なし           | なし  | 不要 |



## 付録B ユーザー設定

### B.1. ユーザー設定

**Teiid Designer** では、さまざまなモデリングおよび UI の動作をカスタマイズできるオプションまたは設定を使用できます。設定にアクセスするにはメインツールバーで **Window > Preferences** と選択します。



図B.1 Preferences ダイアログ

### B.2. Teiid Designer の設定

通常の **Teiid Designer** の設定には以下が含まれます。

- ※ **Enable auto-creating of a source model's data source on Server** - デフォルト名に一致するデータソースが Teiid サーバーに存在しない場合、これを自動作成するかどうかを指定します。
- ※ **Default Teiid Server Version** - Servers View でサーバーが定義されていない場合、ターゲットとするサーバーモデリングのバージョンを決定します。つまり、サーバーの定義および接続なしでモデルを設計することが可能です。ただし、使用可能な値はインストール済みの teiid ランタイムクライアントに限定されます。

- ※ **Always open editor without prompting** - モデルを変更または編集する場合、モデルを開く必要があります。このボックスを選択すると、ユーザーがモデルを変更するときにモデルがエディターで自動的に開かれます。ボックスにチェックマークを入れないと、操作が完了する前にエディターが開かれることが通知されます。
- ※ **Open Designer perspective when model is opened** - モデルがインポートしたプロジェクトによって開かれ、**New > Teiid Metadata Model** メニューと Teiid Designer パースペクティブが開けない場合、パースペクティブを自動的に開き、モデルの作業を行いたいことがあります。これには 3 つの設定があります。Always open を選択すると確認せずに常にパースペクティブが開かれます。Never open を選択すると、Teiid Designer パースペクティブが開かれませんが、Prompt を選択すると、Teiid Designer パースペクティブを開くかどうかを常に要求されます。
- ※ **Check and update imports during save** - 場合によっては、モデルの編集時に他のモデルにあるオブジェクトを参照するモデルのオブジェクトを追加または削除することがあります。モデルインポートは、各モデル内でこれらの依存関係を追跡します。ビルド中に検証エラーや警告が発生することがあります。このボックスを選択すると、保存処理中にインポートを自動的にチェックおよび更新します。これにより、モデルから不要なインポートが削除され、必要なインポートがモデルに追加されます。このボックスを選択しないと、インポートの更新は実行されません。

### B.3. ダイアグラムの設定

複数のダイアグラム設定を使用し、ダイアグラムをカスタマイズできます。

- ※ **Routers** - パッケージおよびカスタムダイアグラムの関係リンク (外部キー - プライマリーキーの関係) はカスタマイズできます。利用できるオプションには、Orthogonal (デフォルト)、Direct、Manual (ユーザー定義のブレイクポイント) が含まれます。
- ※ **Font Settings** - フォントタイプ、スタイル、およびサイズを選択します。
- ※ **Background Color Settings** - 各ダイアグラムタイプの背景色を選択し、タイプを区別できるようにします。
- ※ **Model Size** - 大きなダイアグラムを表示するには大変時間かかることがあります。この設定を使用すると、ダイアグラムで表示できるオブジェクトの最大数を設定できます。この数を超えると、警告が表示され、ダイアグラムが構築されません。
- ※ **Relationship Options** - UML タイプの関係は複数の方法でカスタマイズできます。Role Names および Multiplicity ラベルを表示または非表示にするには、**Show Role Names** および **Show Multiplicity** というラベルが付けられたチェックボックスを使用します。

**Diagram** ← ▾ → ▾ ▾

General settings for Diagrams

**Notations**

Default Notation:  ▾

**Routers**

Default Router Style:  ▾

**Font Settings**

Font Type: Sans - Regular - 8

**Background Color Settings**

|                         |   |                              |   |
|-------------------------|---|------------------------------|---|
| Dependency Diagram:     | <input type="text" value="Light Blue"/> | Custom Diagram:              | <input type="text" value="Light Blue"/> |
| Mapping Diagram:        | <input type="text" value="Light Blue"/> | Relationship Diagram:        | <input type="text" value="Teal"/>       |
| Transformation Diagram: | <input type="text" value="Light Blue"/> | Custom Relationship Diagram: | <input type="text" value="Light Blue"/> |
| Package Diagram:        | <input type="text" value="Light Blue"/> |                              |   |

**Model Size**

Warn If Model Size is Larger Than

**Relationship Options**

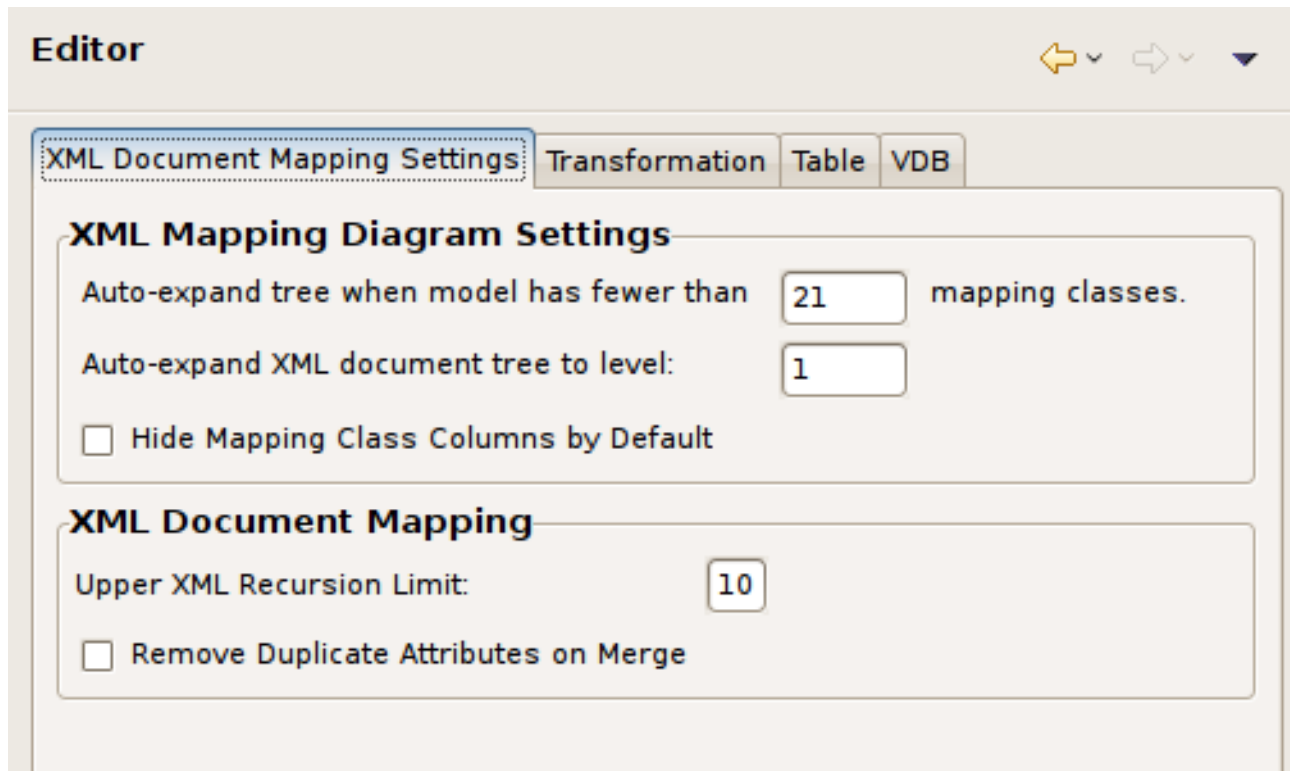
- Show Role Names
- Show Multiplicity

図B.2 ダイアグラム設定パネル

## B.4. エディターの設定

### B.4.1. XML ドキュメントの設定

XML Document Mapping Preferences で、Mapping Diagram と Recursion Editor (XML) の動作をカスタマイズできます。



図B.3 XML ドキュメント設定パネル

### B.4.2. Table Editor の設定

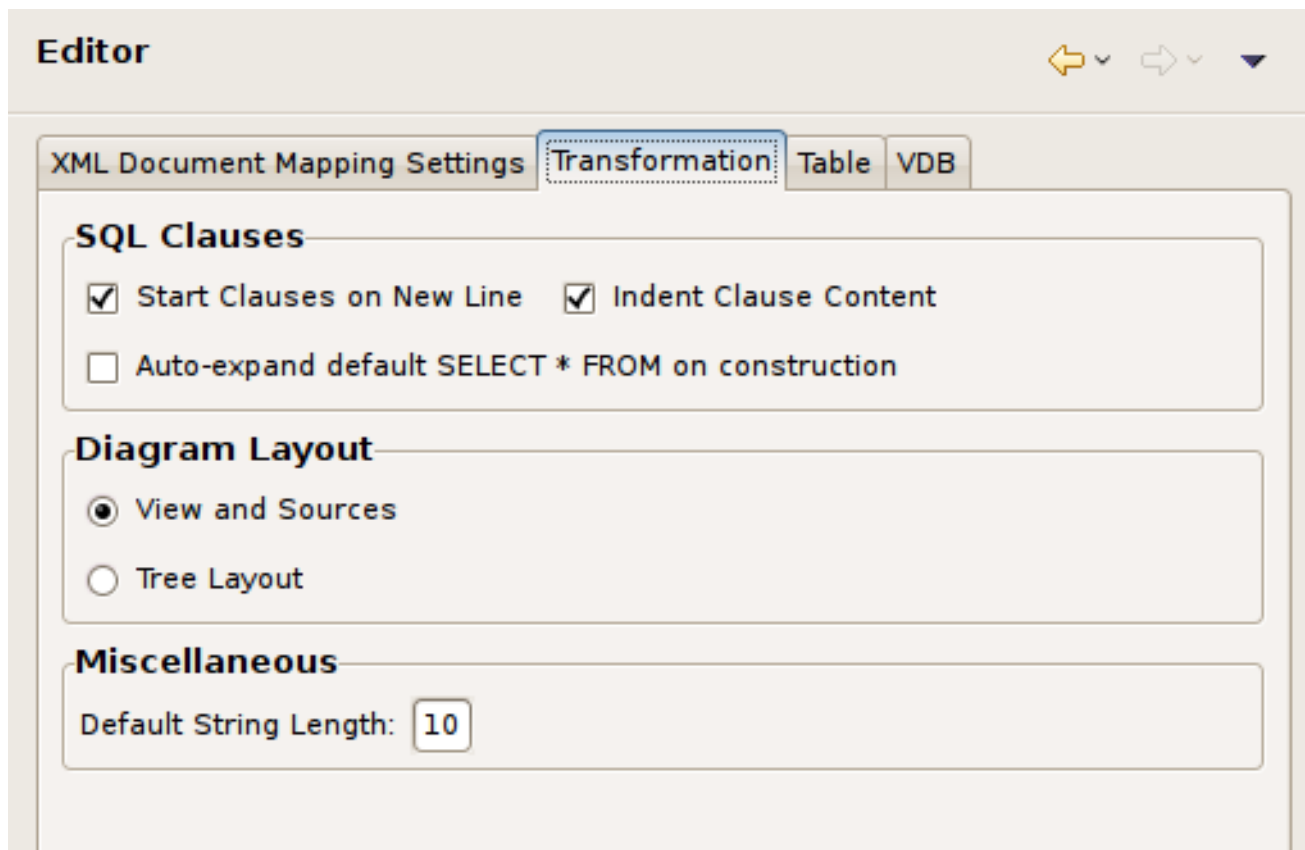
**Table Editor Preferences** では、各モデルオブジェクトタイプの順番と情報内容をカスタマイズできます。



図B.4 Table Editor 設定パネル

### B.4.3. Transformation Editor の設定

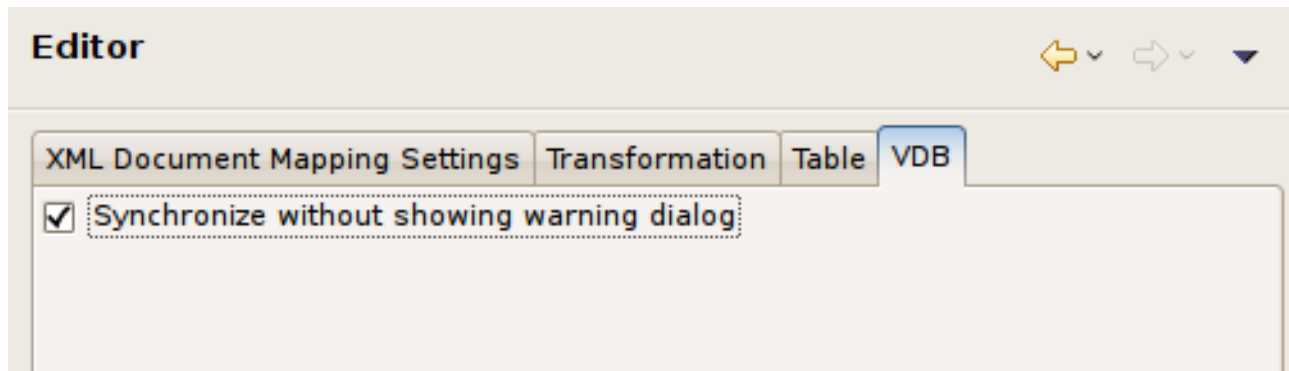
**Transformation Editor Preferences** では、SQL フォーマット、ダイアグラムレイアウト、およびデフォルトのビューエンティティプロパティをカスタマイズできます。



図B.5 Transformation Editor 設定パネル

#### B.4.4. VDB Editor の設定

VDB Editor Preferences では、VDB Editor の動作をカスタマイズできます。

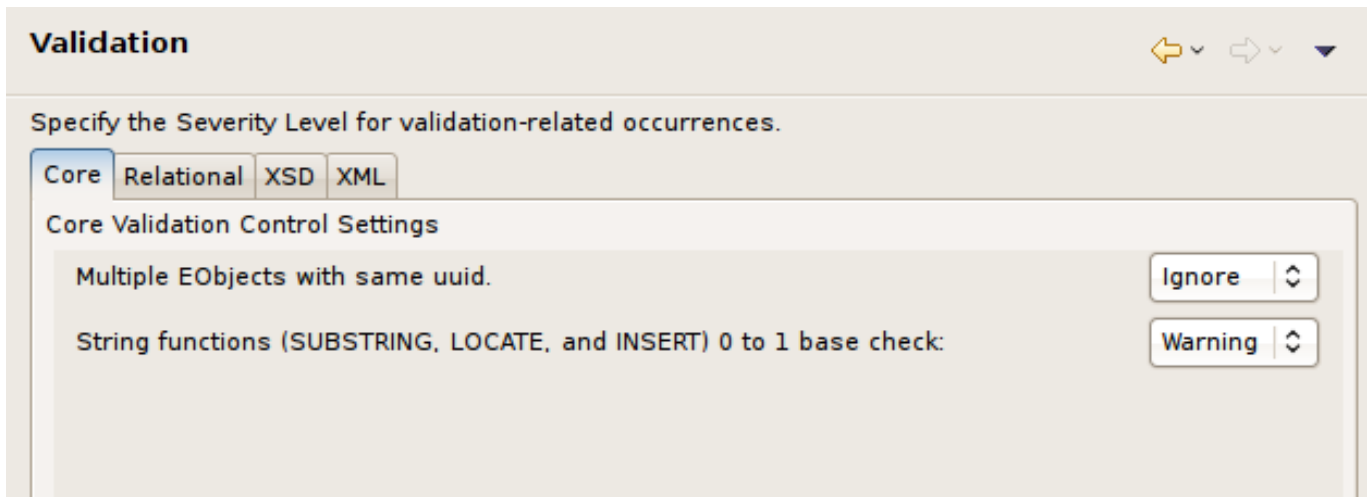


図B.6 VDB Editor 設定パネル

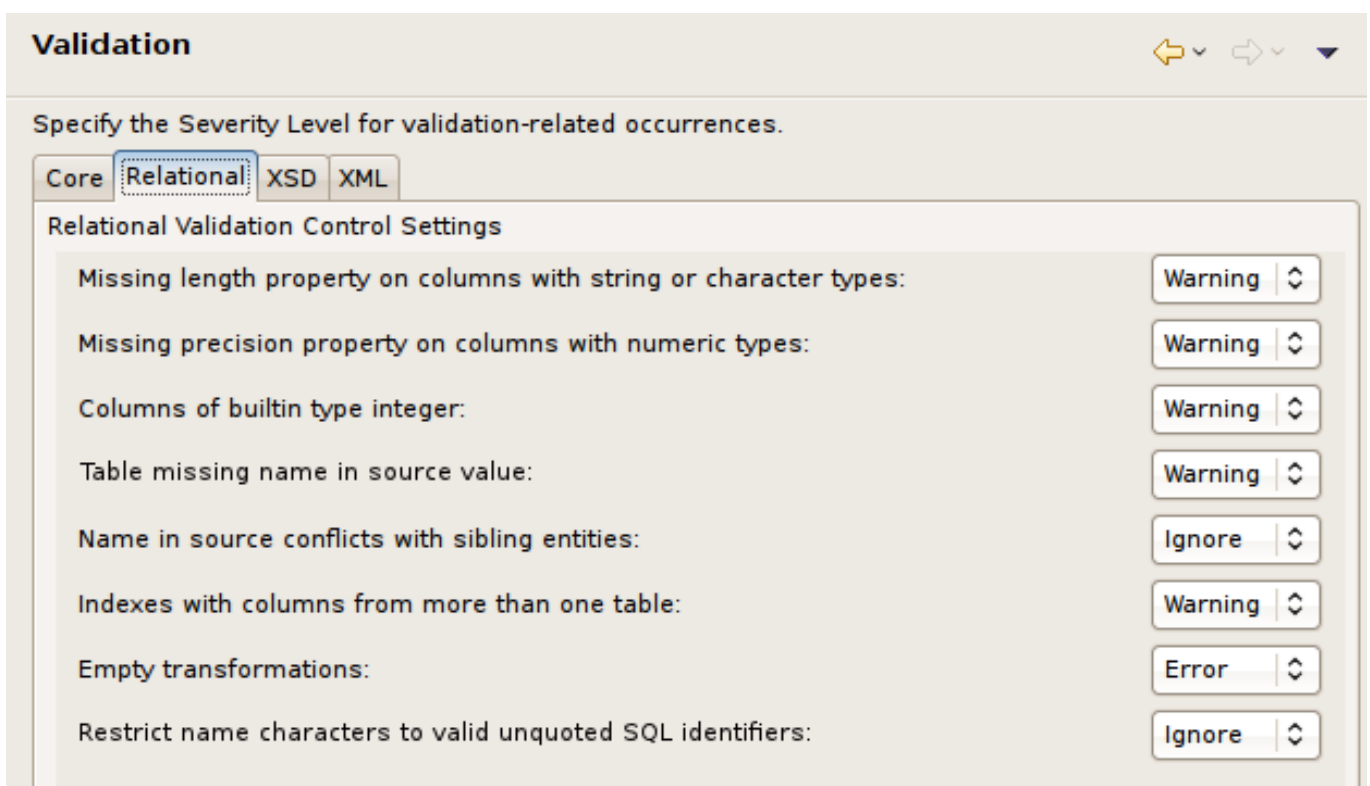
### B.5. 検証の設定

Validation Preferences は、モデル検証中にチェックされる一部のルールの重大度をカスタマイズできます。

以下の検証設定ページには、Core、Relational、XML、および XSD (XML スキーマ) モデルの検証設定が含まれます。



図B.7 Core モデル検証設定パネル



図B.8 Relational モデル検証設定パネル

**Validation** ← ▾ → ▾ ▾

Specify the Severity Level for validation-related occurrences.

Core Relational XSD **XML**

XML Validation Control Settings

|  |           |
|--|-----------|
| XML Document Elements/Attributes not referencing an XML Schema component:  | Warning ▾ |
| Excluded Element from XML Document required by XML Schema:                 | Warning ▾ |
| XML document entity violates max occurs specified by its schema component: | Warning ▾ |
| Unmapped required XML element or attribute:                                | Error ▾   |
| Excluded Elements/Attributes from XML Document are mapped:                 | Warning ▾ |
| Mapped XML Element/Attribute has fixed or default value:                   | Warning ▾ |
| Mapped XML Elements/Attributes with zero minimum occurrences:              | Ignore ▾  |
| Mapped XML Elements/Attributes with one maximum occurrence:                | Warning ▾ |
| XML Root Element mapped to Mapping Class:                                  | Warning ▾ |
| Mapped XML Elements/Attributes Nillable:                                   | Ignore ▾  |
| Incompatible Datatypes for Column-to-Element/Attribute Mappings:           | Warning ▾ |

Restore Defaults Apply

図B.9 XML ドキュメントモデル検証設定パネル

**Validation** ← ▾ → ▾ ▾

Specify the Severity Level for validation-related occurrences.

Core Relational **XSD** XML

XSD Validation Control Settings

|   |          |
|---|----------|
| XML Schema Document validation problems | Ignore ▾ |
|---|----------|

図B.10 XSD スキーマモデル検証設定パネル



### 注記

エラーに対する重大度レベルを大きくすると、検証中に設定違反が見つかった場合に VDB のテストまたは Web サービスのデプロイができなくなります。



## 付録C Teiid Designer UI 参照

### C.1. Teiid Designer パースペクティブ

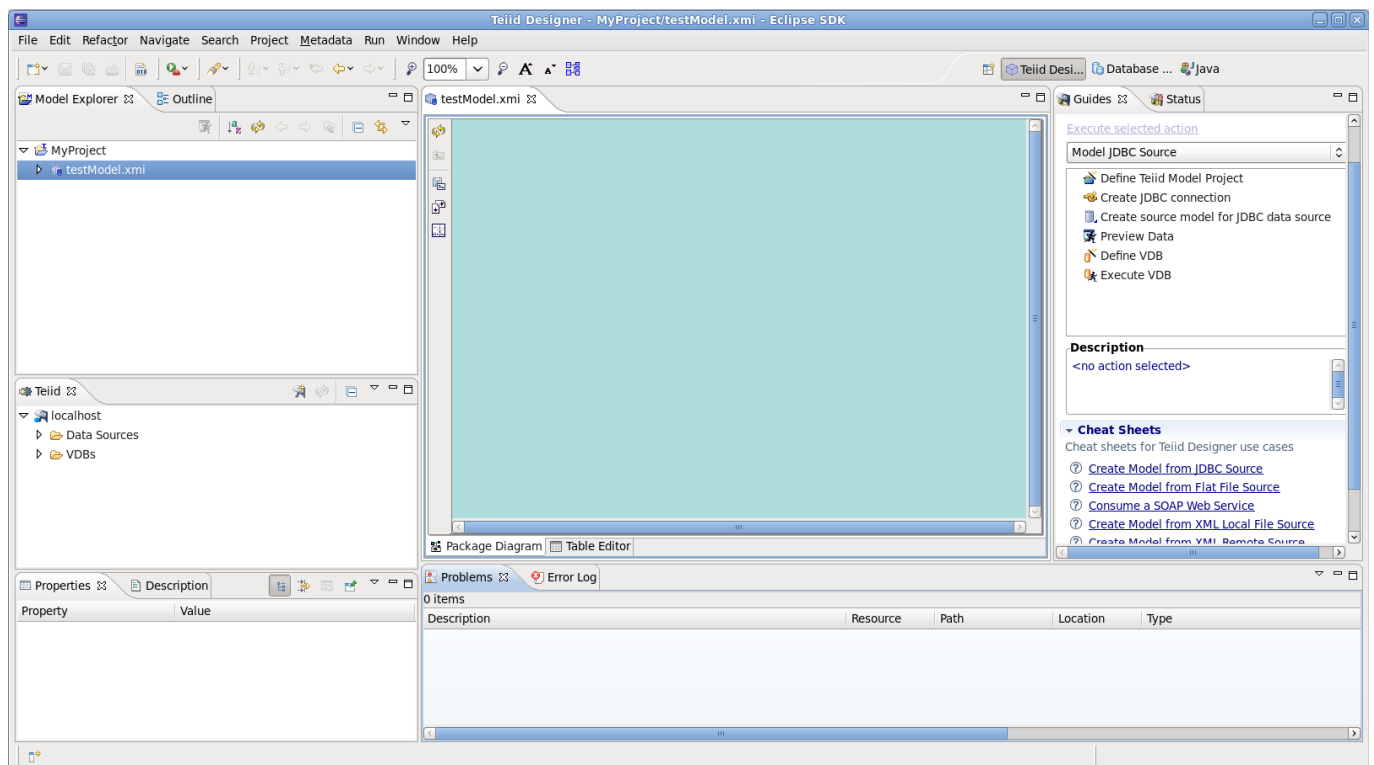
**Teiid Designer** は、パースペクティブから視覚的なレイアウトを制御する [Eclipse Workbench](#) 環境を使用します。パースペクティブはビューおよびエディターの最初のセットおよびレイアウトを定義します。各パースペクティブはアプリケーションウィンドウ内で同じエディターのセットを共有します。各パースペクティブは特定のタスクセットを完了するための機能セットを提供します。

パースペクティブは、特定のメニューおよびツールバーの表示も制御します。パースペクティブは、パースペクティブをカスタマイズするために変更できる表示可能なアクションセットを定義します。構築したパースペクティブをこのように保存し、後で開くことができる独自のカスタムパースペクティブを作成できます。

### C.2. Teiid Designer パースペクティブ

**Teiid Designer** パースペクティブでは、基礎的なモデル編集および管理機能にアクセスできます。このパースペクティブには、以下の主な UI コンポーネントとコンポーネントのグループが含まれます。


- Model Explorer ビュー - モデルオブジェクトの Teiid ツリービュー。
- Server ビュー - JBoss Data Virtualization Server インスタンスのビュー。インストールされた Teiid ランタイムの接続されたインスタンスに対してコンテンツのビューを提供します。
- Model Editor - `.xmi` メタデータモデルファイルを対象とするカスタムエディター。
- Properties ビュー - 選択されたワークベンチオブジェクトの標準的なプロパティ値。
- Guides ビュー - 使いやすさを向上する Guides および Status ビュー。
- Miscellaneous ビュー - Problems ビュー、Message Log ビュー、および SQL Results ビュー (Preview Data が実行された場合に開かれる) が含まれます。



## 図C.1 Teiid Designer パースペクティブレイアウト

### C.3. パースペクティブを開く

パースペクティブを開く方法は 2 つあります。

- ※ Using the **Open Perspective** button  on the shortcut bar.
- ※ **Window > Open Perspective** メニューからパースペクティブを選択します。

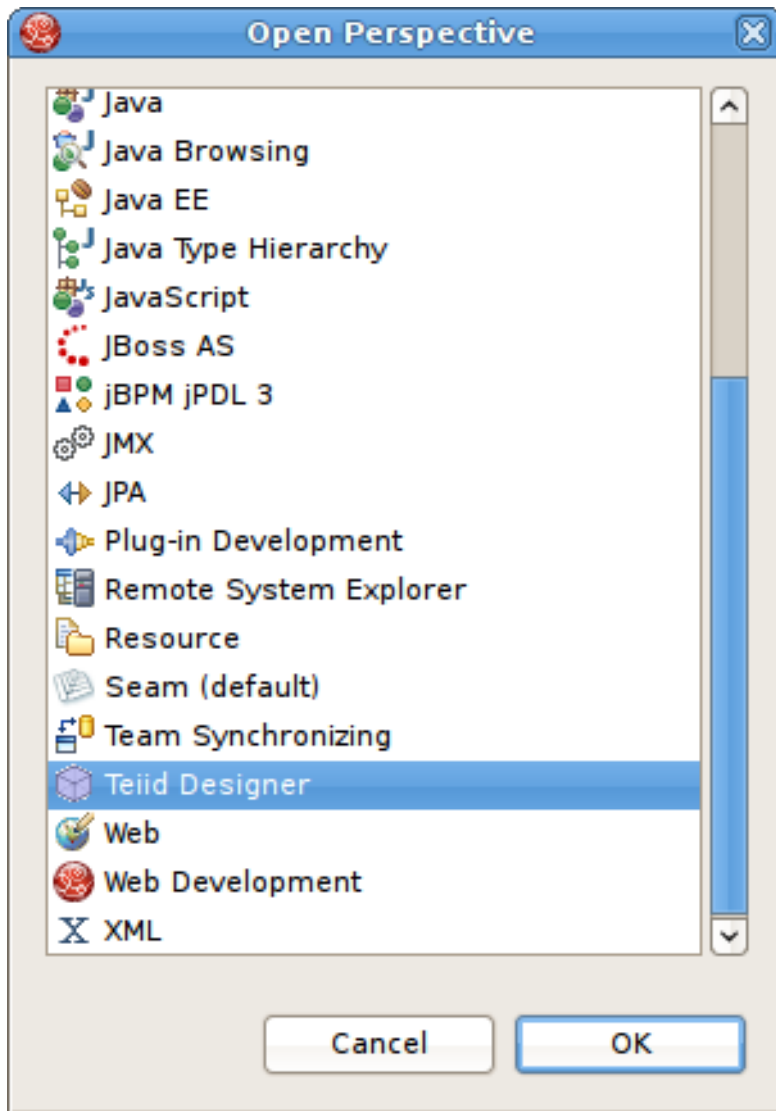
ショートカットバーのボタンを使用してパースペクティブを開くには、以下を実行します。

1. Click the **Open Perspective** button .
2. メニューが表示され、**Window > Open Perspective** メニューと同じ選択肢が表示されます。メニューで **Other** をクリックします。



図C.2 パースペクティブメニュー

3. **Select Perspective** ダイアログで **Teiid Designer** を選択し、**OK** をクリックします。

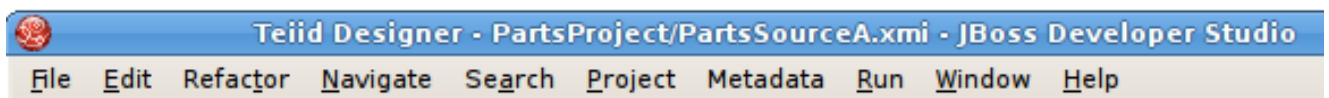


図C.3 Select Perspective ダイアログ

Teiid Designer パースペクティブが表示されます。

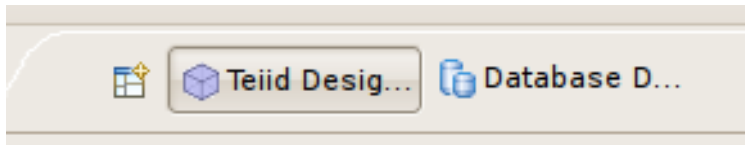
パースペクティブの追加機能を確認してください。

- ※ 使用中のパースペクティブがウィンドウのタイトルになります。



図C.4 ワークベンチウィンドウタイトルバー

- ※ ショートカットバーには複数のパースペクティブが含まれることがあります。押された状態のパースペクティブボタンが現在のパースペクティブを示します。
- ※ パースペクティブのフルネームを表示するには、パースペクティブバーを右クリックして **Show Text** をクリックします。アイコンのみを表示するには **Hide Text** をクリックします。
- ※ 開かれたパースペクティブを即座に切り替えるには、パースペクティブボタンを選択します。各パースペクティブによってビューのセットが異なることに注意してください。



図C.5 ワークベンチウィンドウタイトルバー

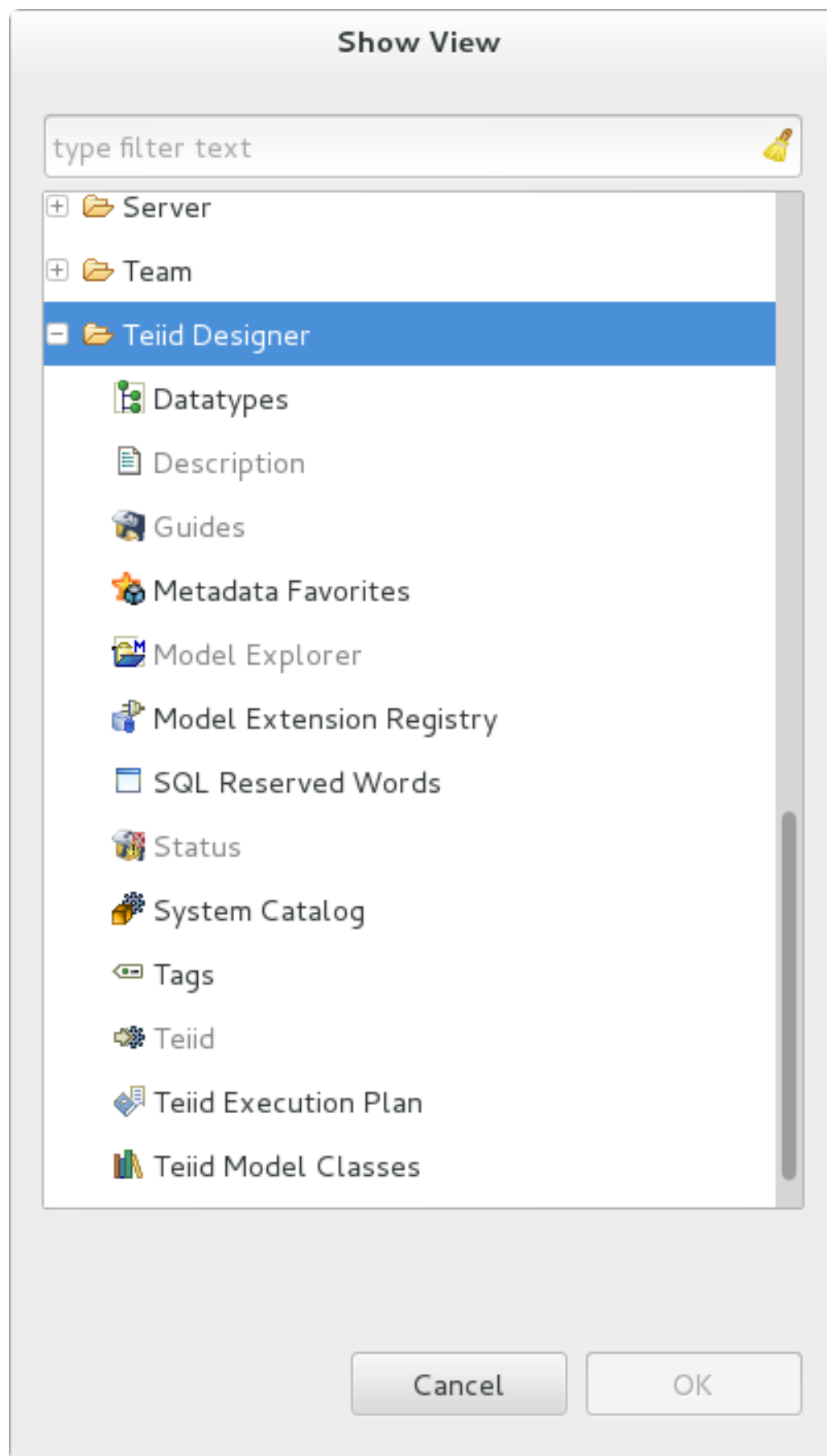
## C.4. 詳細情報

パースペクティブ、ビュー、および Eclipse ワークベンチの詳細は、公式の [Eclipse ドキュメント](#) を参照してください。

## 付録D Teiid Designer ビュー

### D.1. Teiid Designer ビュー

ビューは、モデルまたはモデリングセッションからのデータをさまざまな形式で提供するドッキング可能なウィンドウです。ビューによってはワークスペースの選択に応じて特定のモデルエディターやその内容をサポートします。ここでは、**Teiid Designer** で使用されるほとんどのビューについて取り上げます。完全なリストは、**Teiid Designer** カテゴリ下にあるメインメニューの **Window > Show View > Other...** ダイアログを参照してください。

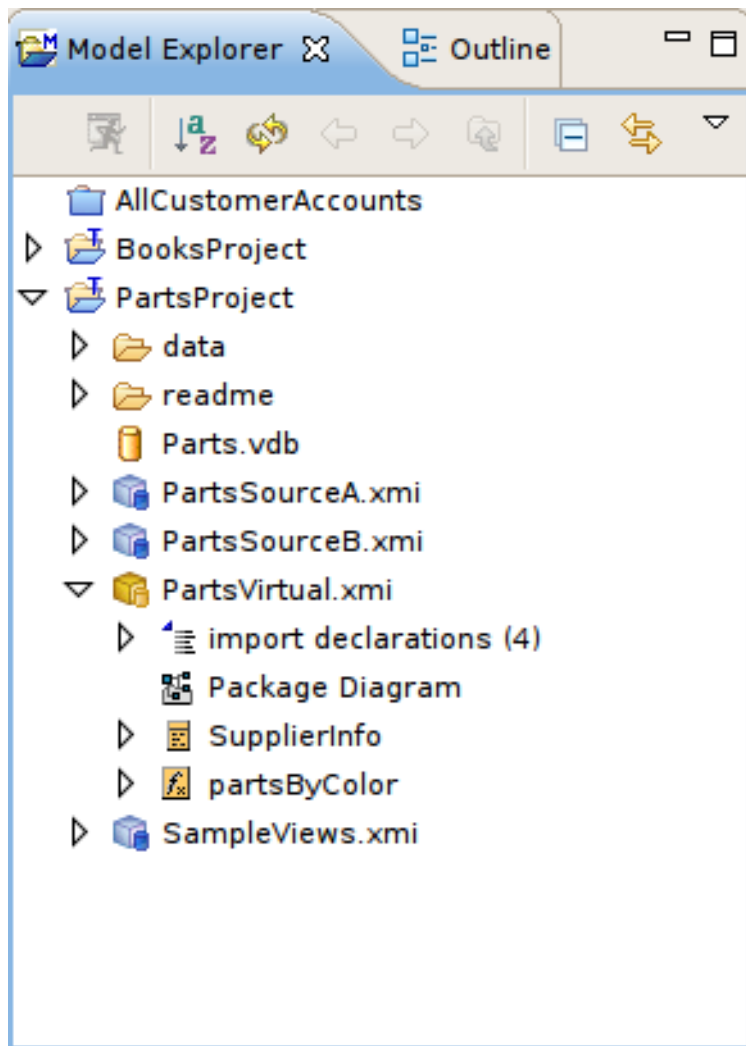


図D.1 Show View ダイアログ

## D.2. Model Explorer ビュー








**Teiid Designer** では、複数のモデルと対応するリソースまたは依存するリソースが含まれる複数のプロジェクトを管理できます。**Model Explorer** は、これらのリソースの簡単なファイル構造のビューを提供します。



**Model Explorer** は次のようにツールバーとツリービューで構成されます。



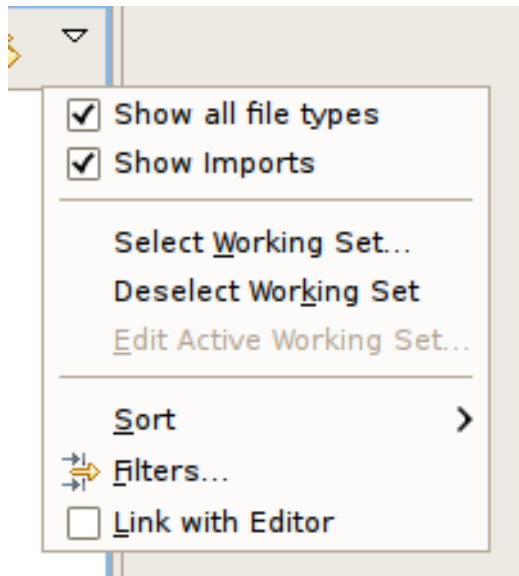
図D.2 Model Explorer ビュー

ツールバーは 9 つのアクションで構成されます。

- ✧  Preview Data - Executes a simple preview query (**SELECT \* FROM** ).
- ✧  Sort Model Contents - Sorts the contents of the models based on object type and alphabetizing.
- ✧  Refresh Markers - Refreshes error and warning markers for objects in tree.
- ✧  Back - Displays the last Go Into location. (See Eclipse Help)
- ✧  Forward - Displays the next Go Into location. (See Eclipse Help)
- ✧  Up - Navigates up one folder/container location. (See Eclipse Help)
- ✧  Collapse All - Collapses all projects.

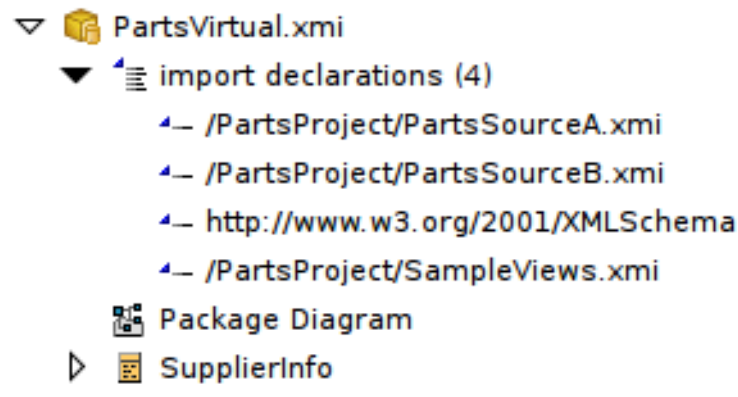
- ※  Link with Editor - When object is selected in an open editor, this option auto-selects and reveals object in Model Explorer.
- ※  Additional Actions

追加のアクションは以下の図のとおりです。



図D.3 追加のアクション

**Show Model Imports** が選択されていると、インポートは以下のようにモデルリソース直下に表示されます。



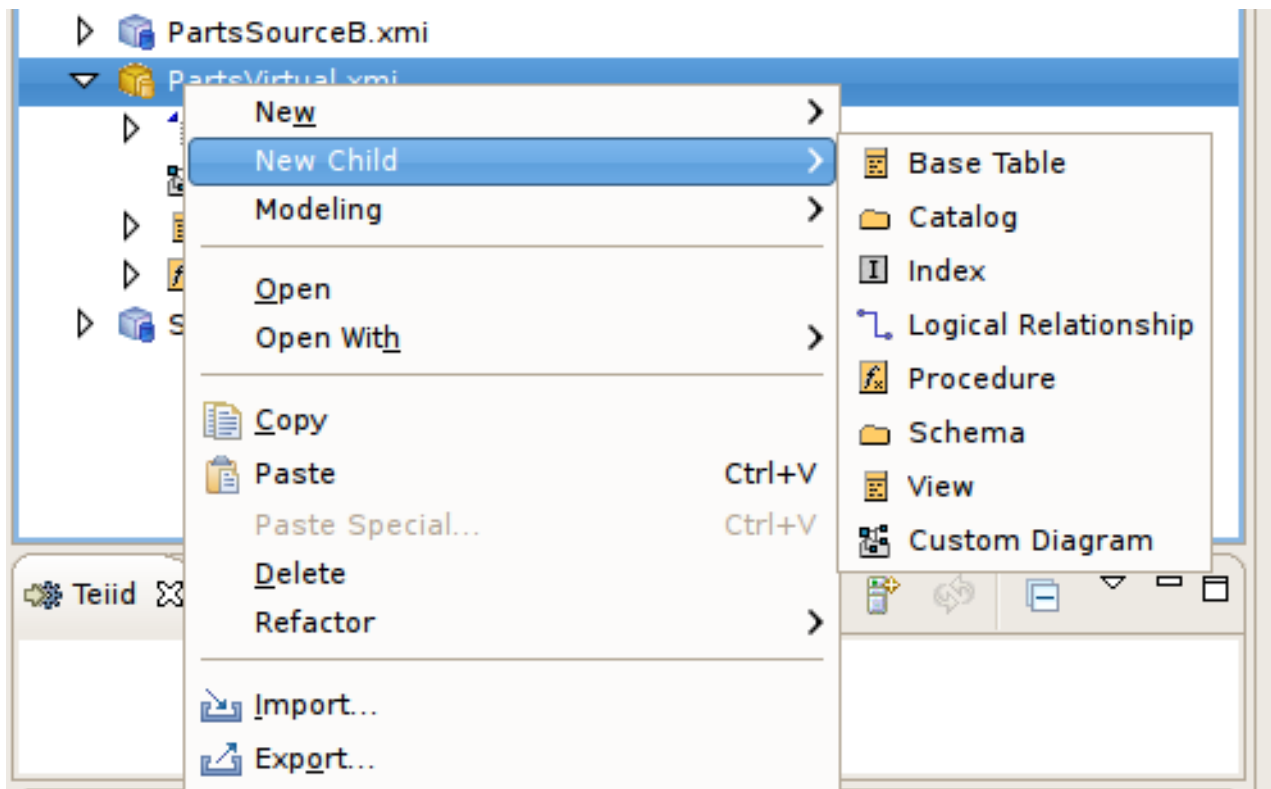
図D.4 Show Model Imports アクション

### D.3. 選択ベースのアクションメニュー

Model Explorer で特定のオブジェクトを選択すると、**Teiid Designer** が利用可能なアクションのカスタマイズされたメニューを表すコンテキストが提供されます。

たとえば、ビューモデルを選択すると、モデルコンテンツの編集を管理するためのハイレベルなオプションが提供されます。さまざまな演算が実行され、**Teiid Designer** で利用できる重要なその他のアクションへ即座にアクセスできます。これには、モデルタイプに基づいた特別なアクションが含まれます。





図D.5 コンテキストメニューの例

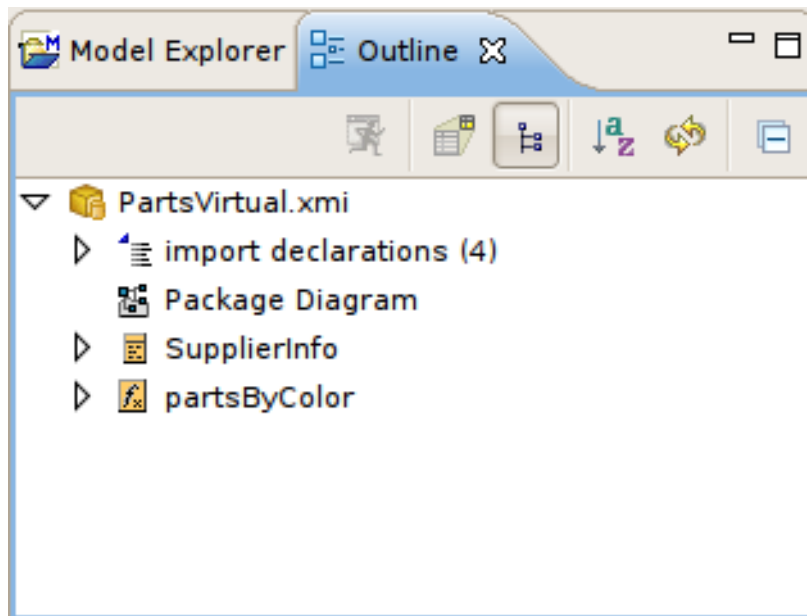
## D.4. Outline ビュー

**Outline View** は、特定モデル専用のツリービュー (エディターで開かれます) と、対応する **Diagram Editor** で開かれるダイアグラムのスケールされたサムネイルダイアグラムを両方提供するユーティリティビューです。

**Outline View** を表示するにはそのタブをクリックします。開かれたエディターがない場合は、ビューに **Outline is not available** というメッセージが表示されます。**Model Editor** が開かれている場合、表示されたツリーのルートは、**Teiid Designer** で現在指定されているエディターのモデルになります (上部のタブ)。

## D.5. オンラインツリービュー


このツリービューは、Model Explorer と同じ基本的な編集および移動動作を提供します。この他に、モデルのオブジェクトの順序や親を変更できるようにするドラッグアンドドロップ機能が追加機能として提供されます。

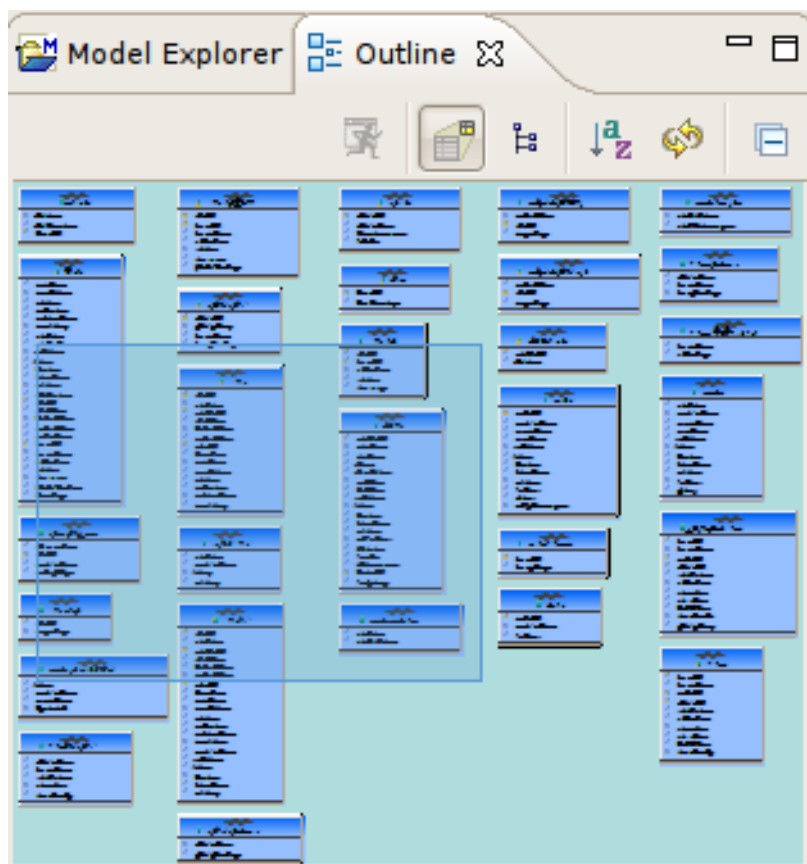


図D.6 Outline ビュー

## D.6. アウトラインサムネイルビュー

The **Outline View** also offers you a way to view a thumbnail sketch of your diagram regardless of its size.

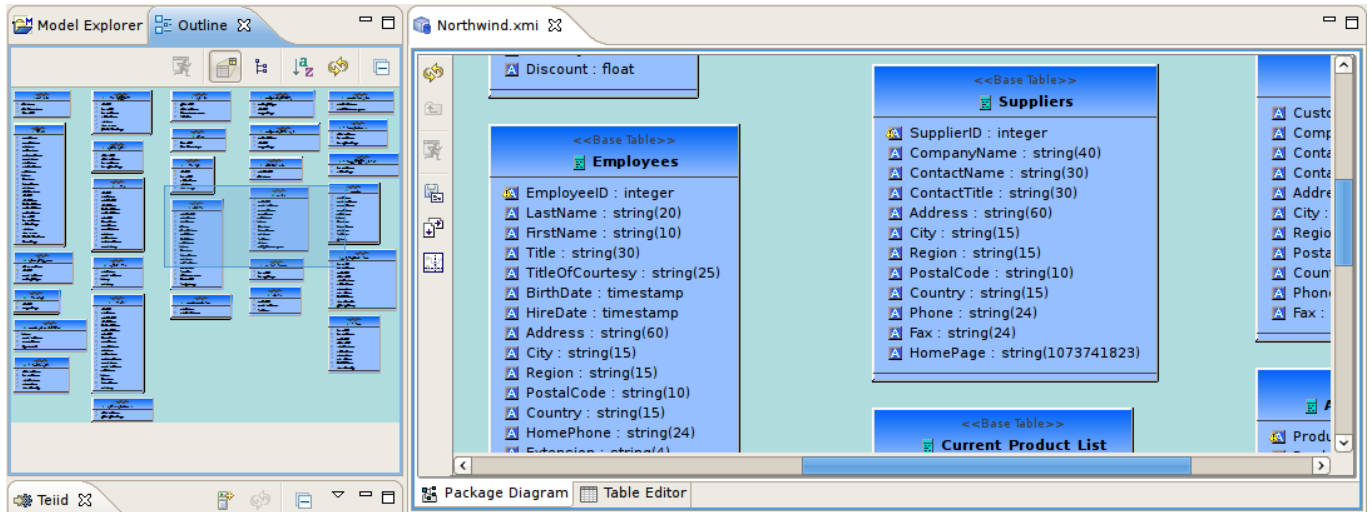
To view this diagram thumbnail from the **Outline** panel, click the **Diagram Overview** button  at the top of the view. The diagram overview displays in the **Outline View**.



図D.7 Outline ビュー

ビューにはダイアグラム全体のサムネイルが含まれます。網掛けの部分が **Diagram Editor** ビューで表示される部分です。

ダイアグラムの特定の部分を移動するには、網掛けの部分をクリックし、**Diagram Editor** で表示したい場所にドラッグします。

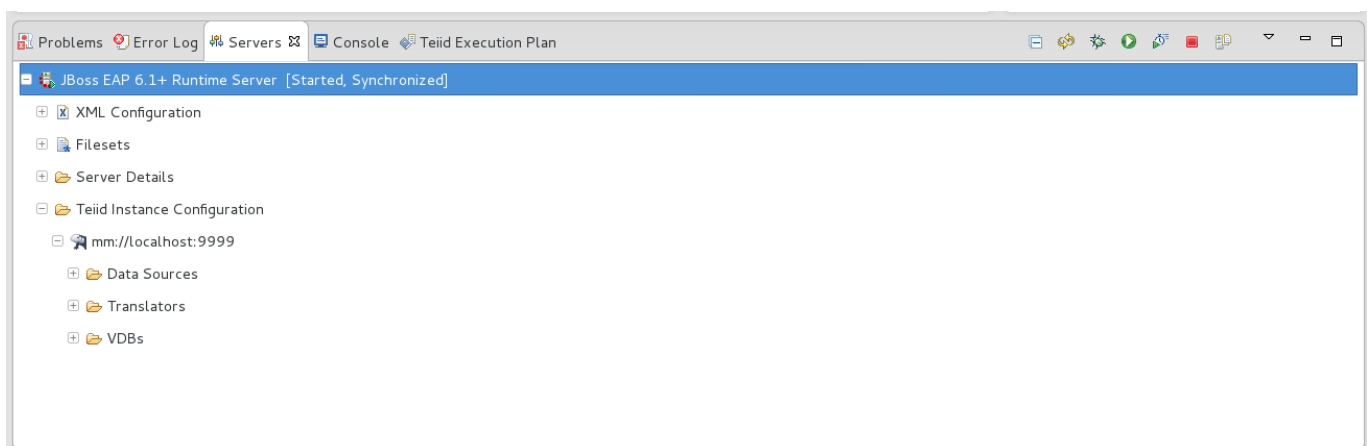


図D.8

## D.7. サーバービュー

Server View は、JBoss Developer Studio 環境内でサーバーインスタンスとそれらのコンテンツを表示および管理する方法を提供します。Teiid Designer は JBoss Data Virtualization サーバーの一部としてインストールされるため、そのコンテンツも親の JBoss の一部として表示されます。

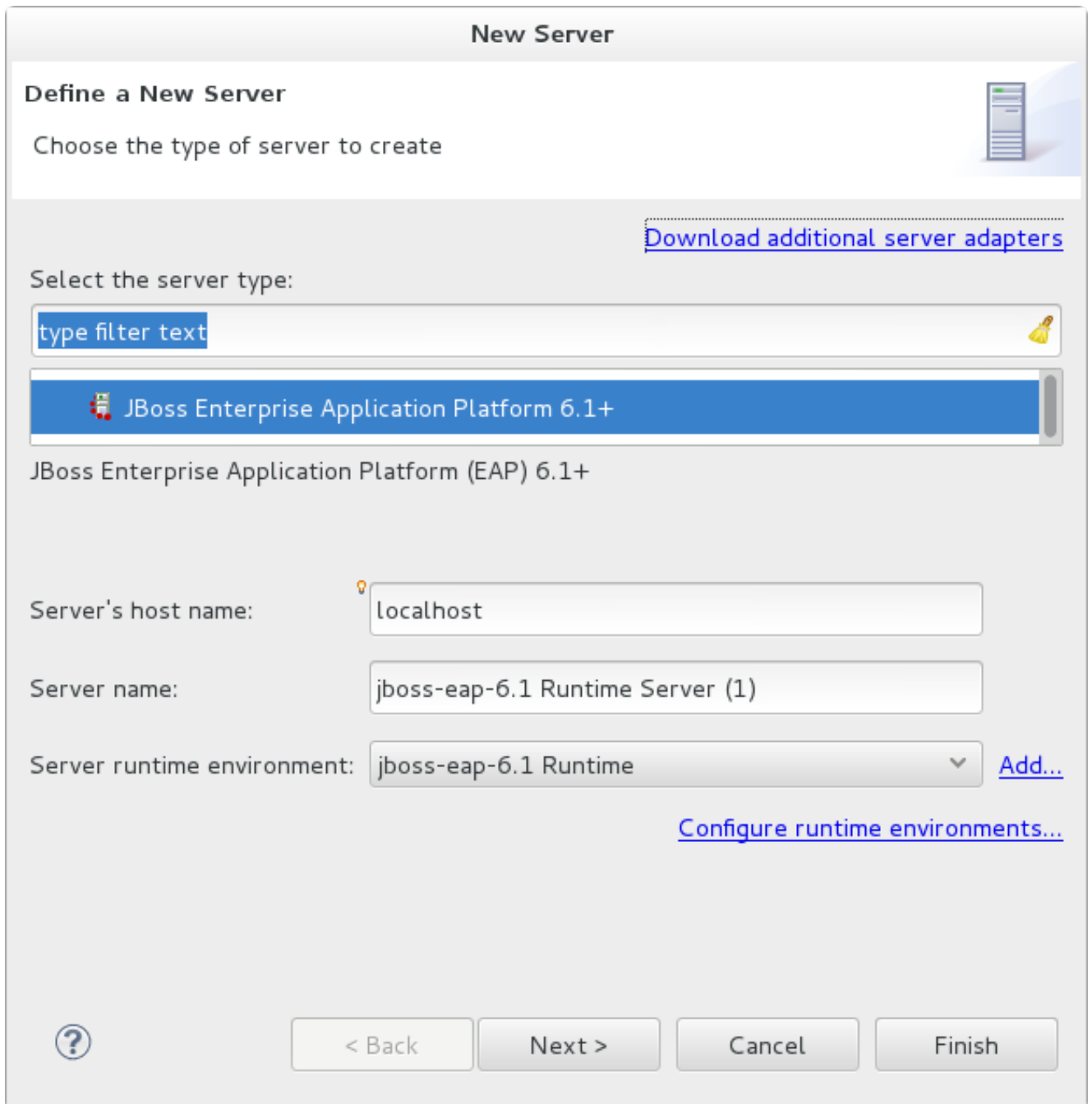
Server View を表示するには、**Window > Show View > Other...** とクリックし、**Show View** ダイアログを表示します。**Server > Servers** と選択し、**OK** をクリックします。



図D.9 サーバービュー

Teiid インスタンスを作成するには、以下を行います。

- ※ Server View で **New...** を選択すると、JBoss Data Virtualization Server ウィザードが起動されます。JBoss Data Virtualization Server インスタンスを Teiid がインストールされた状態で設定すると、Teiid サーバーへの接続が可能になります。
- ※ **New Server** ウィザードで、JBoss Enterprise Middleware カテゴリー下にある JBoss Enterprise Application Platform サーバータイプを選択し、**Next >** をクリックします。




図D.10 New Server ダイアログ

- \* JBoss Runtime のページで上部の **Browse...** ボタンをクリックし、JBoss Enterprise Application Platform サーバーのインストールフォルダーを選択します。

**New Server Runtime Environment**

**JBoss Runtime**  
JBoss Enterprise Application Platform 6.1+




A JBoss Server runtime references a JBoss installation directory.  
It can be used to set up classpaths for projects which depend on this runtime,  
as well as by a "server" which will be able to start and stop instances of JBoss.

Name

Home Directory [Download and install runtime...](#)

JRE

Configuration file:



図D.11 JBoss Runtime の定義

- ※ 下部の **Browse...** ボタンをクリックし、ファイルシステムの `EAP_HOME/standalone/configuration/` フォルダ下にある `standalone.xml` 設定ファイルを選択します。**Finish** をクリックします。

| Name                             | Size      | Modified           |
|----------------------------------|-----------|--------------------|
| standalone_xml_history           |           | Yesterday at 12:20 |
| teiid-security-users.properties~ | 97 bytes  | 12/18/2013         |
| teiid-security-users.properties  | 96 bytes  | 12/27/2013         |
| teiid-security-roles.properties~ | 195 bytes | 12/27/2013         |
| teiid-security-roles.properties  | 189 bytes | 12/27/2013         |
| standalone-osgi.xml              | 28.7 kB   | 12/24/2013         |
| standalone-ha.xml                | 31.6 kB   | 12/24/2013         |
| standalone-full-ha.xml           | 38.0 kB   | 12/24/2013         |
| standalone-full.xml              | 32.3 kB   | 12/24/2013         |
| <b>standalone.xml</b>            | 27.3 kB   | Monday             |
| modeshape-users.properties       | 85 bytes  | 12/19/2013         |
| modeshape-roles.properties       | 207 bytes | 12/19/2013         |
| mgmt-users.properties            | 876 bytes | 12/24/2013         |
| logging.properties               | 1.9 kB    | 14:58              |
| application-users.properties     | 812 bytes | 08/21/2013         |
| application-roles.properties     | 634 bytes | 08/21/2013         |

図D.12 Teiid 設定ファイルの選択

- ※ 新しい JBoss Data Virtualization サーバー設定が JBoss / Teiid Editor で開かれ、表示されます。このエディターでは、Teiid 管理と JDBC 接続の両方をテストできます。

**JBoss EAP 6.1+ Runtime Server**

### Teiid Instance

▼ Overview

Display Name

Host localhost

Teiid Runtime Version 8.4.0

JBoss Server JBoss EAP 6.1+ Runtime Server

▼ Administration Connection

Administration is performed via the jboss management configuration

Port 9999

[Test Administration Connection](#)

▼ JDBC Connection

User name

Password

Port 31000

JDBC connection uses SSL

[Test JDBC Connection](#)

Overview | Deployment | **Teiid Instance**

Problems | Error Log | **Servers** | Console | Teiid Execution Plan

JBoss EAP 6.1+ Runtime Server [Started, Synchronized]

- + XML Configuration
- + Filesets

### 図D.13 Teiid Editor

このビューで使用できるアクションは次のとおりです

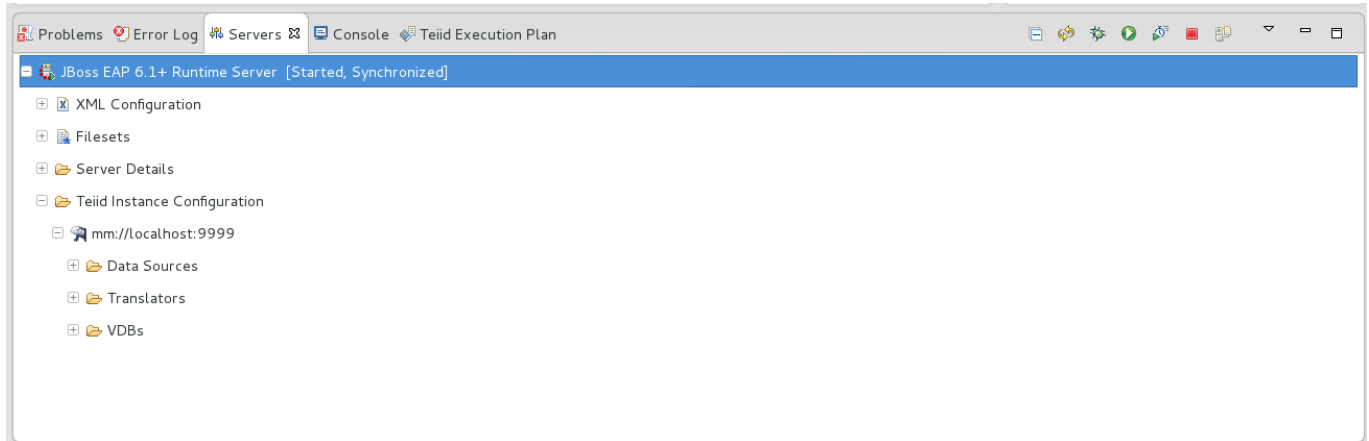
- ✧ Teiid Server Properties - View and edit properties of an existing Teiid instance.
- ✧ Reconnect - Reconnect and refresh contents of the selected Teiid instance.
- ✧ Execute VDB - Creates a JDBC Teiid connection profile and opens the Data Tools Database Development perspective.
- ✧ Undeploy VDB - Removes the selected VDB from the Teiid instance.
- ✧ Create Data Source - Launches the New Data Source wizard.

- ✖ Delete Data Source - Removes the selected Data Source from the Teiid instance.



## 注記

サーバーが起動すると、Server ビューに稼働している Teiid サーバーのデータソース、トランスレーター、および VDB コンテンツが表示されます。



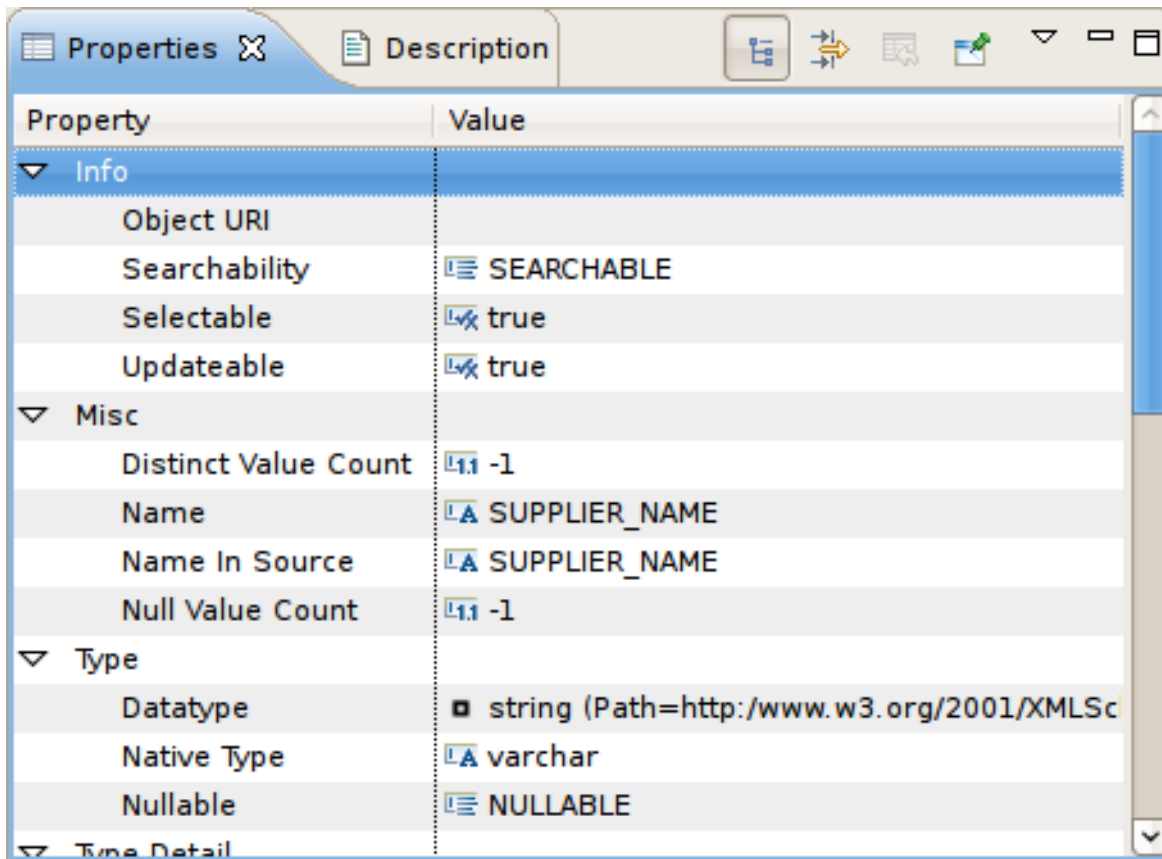
図D.14 Server ビューの Teiid コンテンツ

## D.8. プロパティビュー

**Properties View** は、**Teiid Designer** で現在選択されているオブジェクトの編集機能を提供します。現在指定のビューまたはエディターによって提供される選択によって内容が決定されます。

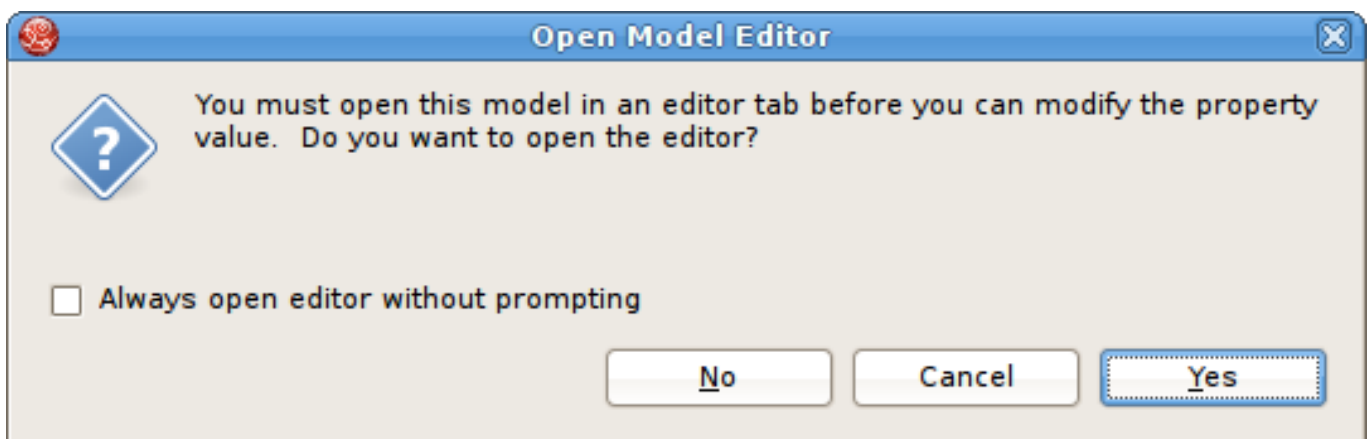
プロパティを編集するには、**Value** 列のセルをクリックします。**Table Editor** と同様に、各セルはプロパティタイプに固有する UI エディターを提供します。





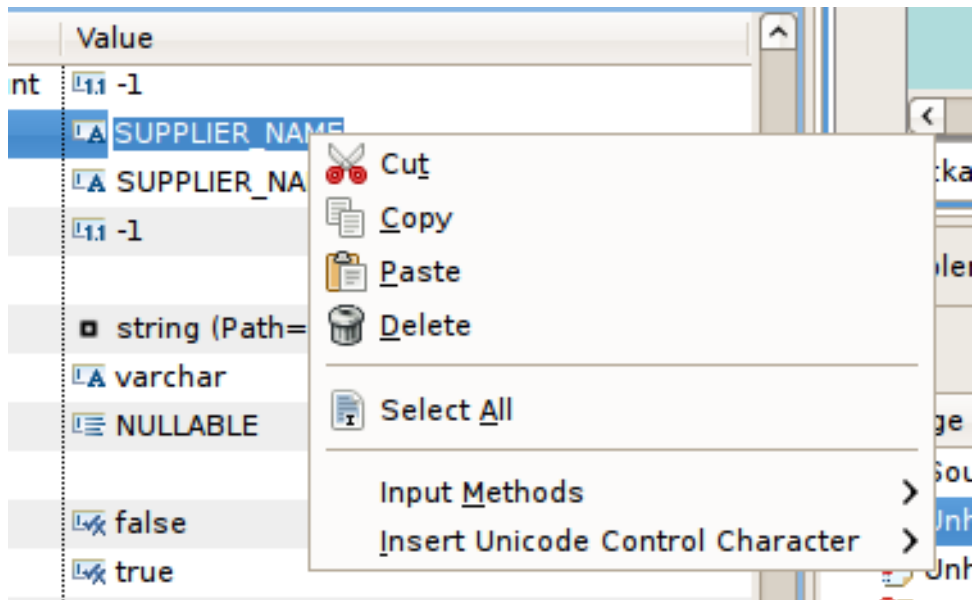
図D.15 プロパティビュー

編集中のオブジェクトのモデルがエディターで開かれていないと、モデルの編集が試行されたことを確認するダイアログが表示されることがあり、ユーザーが確認またはキャンセルするよう要求されます。このダイアログが表示されないようにするには、**Always open editor without prompting**を選択します。Teiid Designer のメイン設定ページでもこのプロパティを選択または未選択することができます。






図D.16 Open Model Editor ダイアログ

以下のように、右クリックで表示されたメニューでプロパティを編集することもできます。



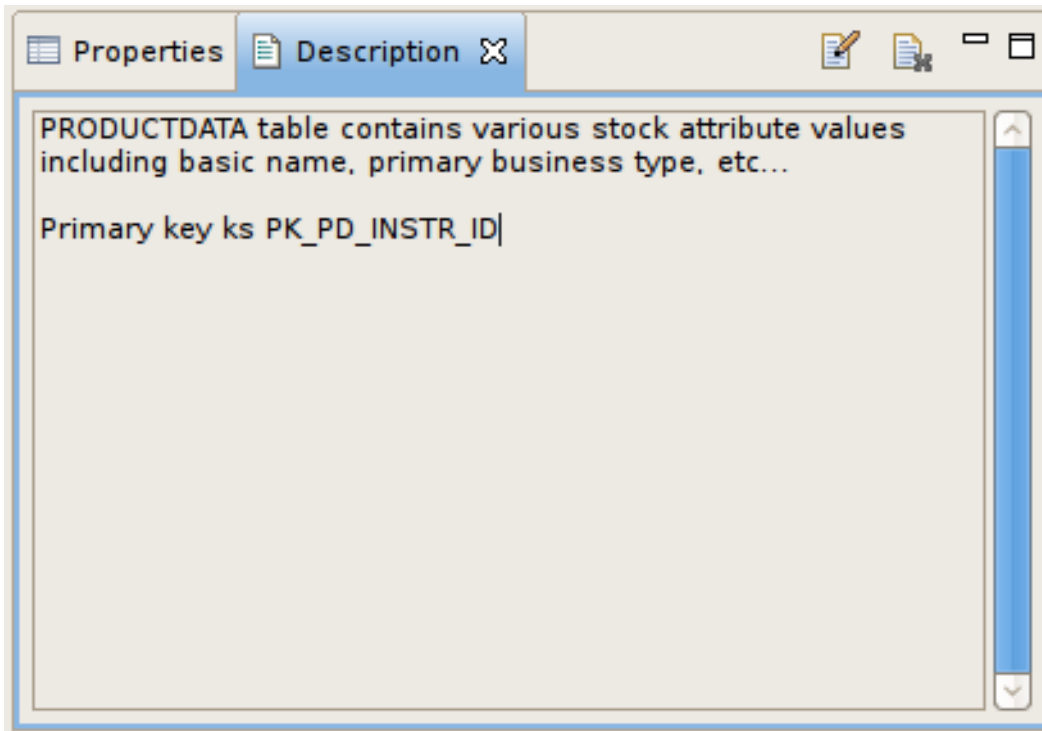
図D.17 Open Model Editor ダイアログ

Properties ツールバーには以下のアクションが含まれます。

- ✧  Show Categories - toggles between categorized properties and flat alphabetical properties list.
- ✧  Show Advanced Properties - shows/hide advanced properties (if available).
- ✧  Restore Default Value - for a selected property, this action will reset the current to a default value (if available).

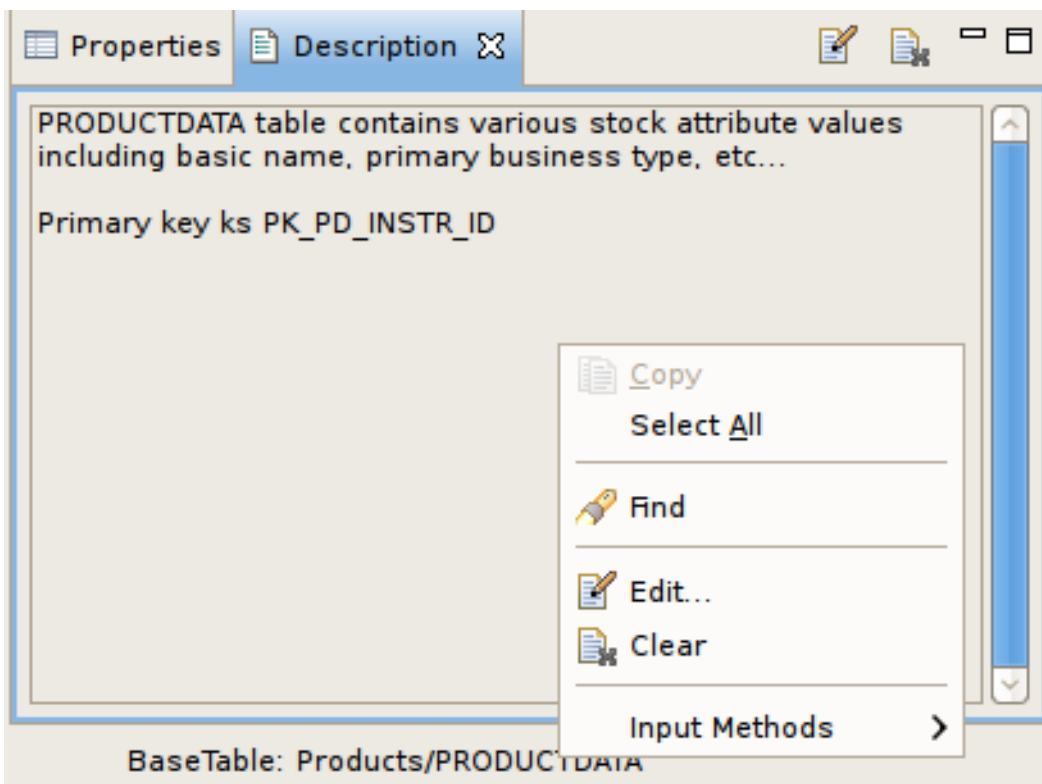
## D.9. 説明ビュー

**Description View** では、モデルまたはモデルオブジェクトの説明を表示および編集 (追加、変更、または削除) できます。**Description View** を表示するには、**Window > Show View > Other...** と選択し、**Eclipse Show View** ダイアログを表示します。**Teiid Designer > Description** ビューをクリックし、**OK** をクリックします。

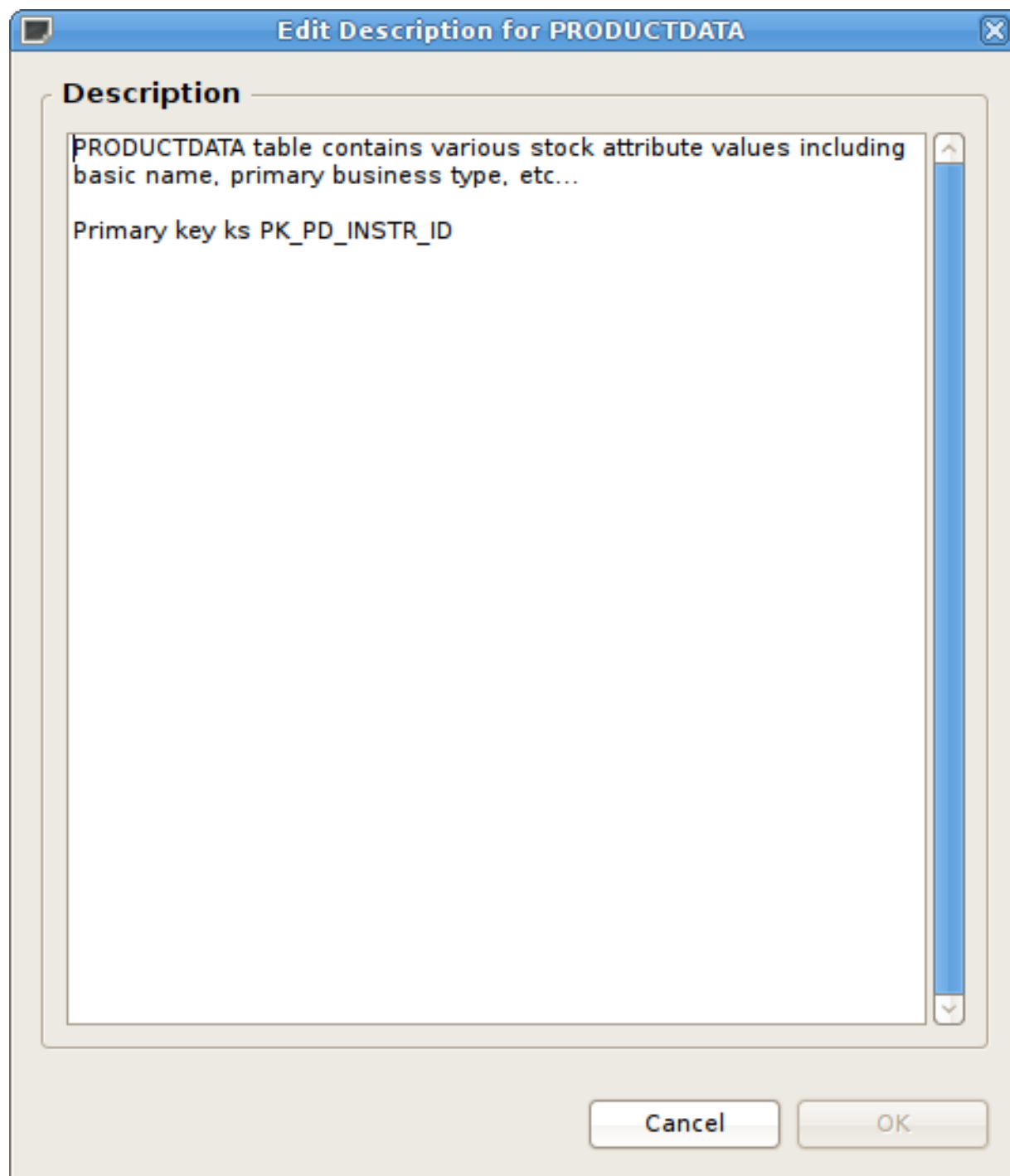


図D.18 説明ビュー

ツールバーで edit description アクションをクリックするか、右クリックしてコンテキストメニューの **Edit** を選択し、**Edit Description** ダイアログを表示します。



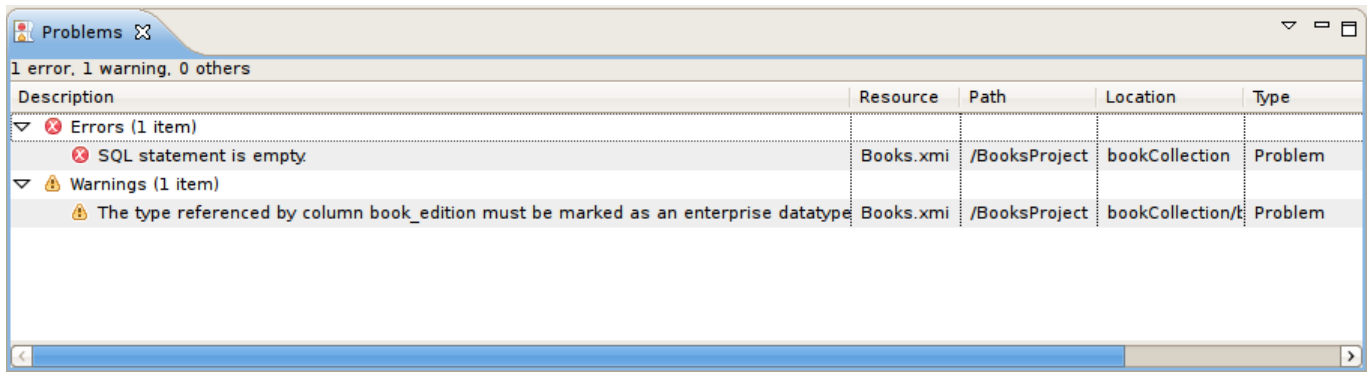
図D.19 Description ビューのコンテキストメニュー



図D.20 Edit Description ダイアログ

## D.10. 問題ビュー




**Problems View** は、ワークスペース内のオープンプロジェクトに含まれるリソースに関連する検証エラー、警告、または情報を表示します。




図D.21 問題ビュー

デフォルトでは、**Problems View** は Teiid Designer パースペクティブに含まれます。**Problems View** に現在のパースペクティブが表示されない場合は **Window > Show View > Other > General > Problems** とクリックします。

ビューのテーブルには 5 つの列があり、以下が含まれます。

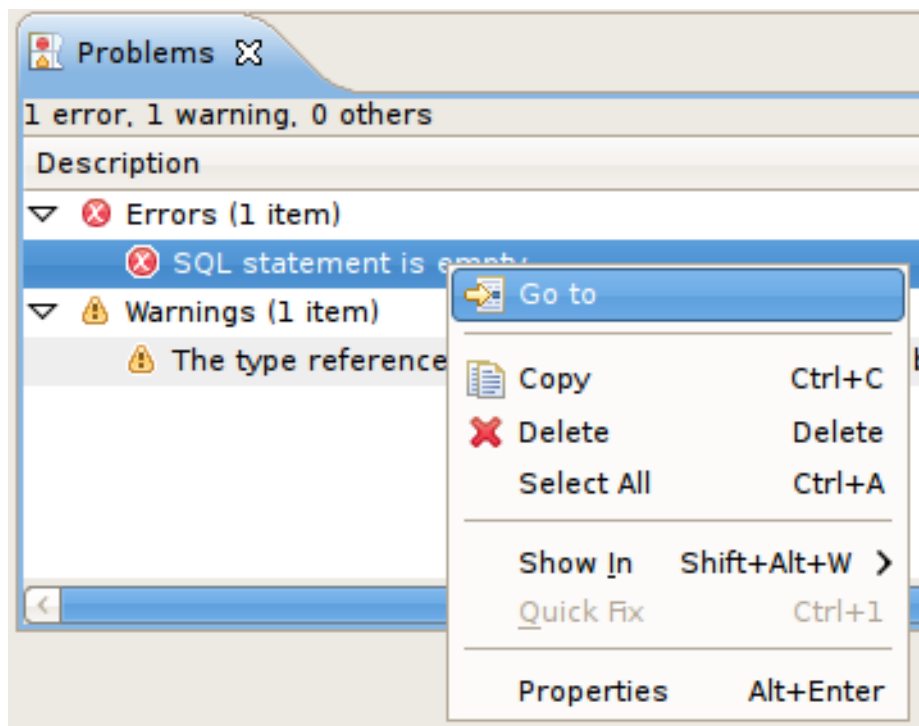
1. Description - A description of the problem preceded by a severity icon (i.e., error , warning , or info ).
2. Resource - リソースの名前。
3. Path - プロジェクト名。
4. Location - 検証エラーがあるリソース内のオブジェクト。
5. type - 検証項目のタイプ。

## D.11. ツールバーメニュー

Click the upside-down triangle  icon to open the View Menu icon to see various options including sorting, filtering, displayed columns and much more.

## D.12. コンテキストメニュー

問題を選択し、右クリックしてコンテキストメニューを開くと、追加のアクションを使用できます。

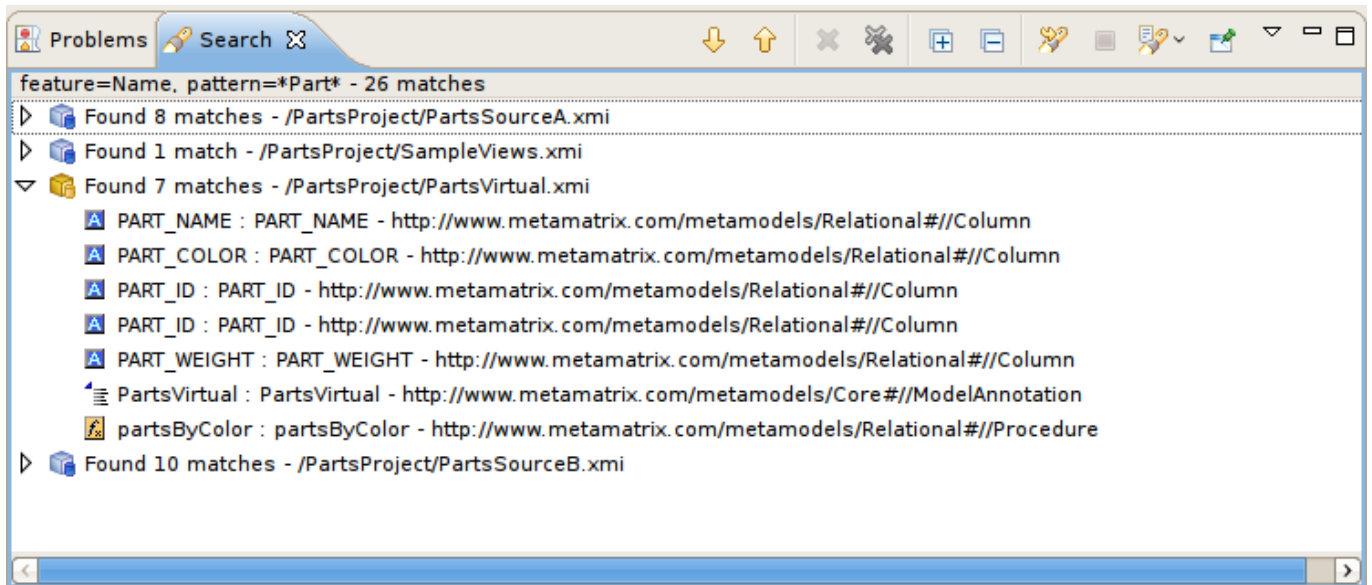


図D.22 Problems View のコンテキストメニュー

- ※ Go To - 適切なエディターを開き、影響を受けたオブジェクトまたは参照されたオブジェクトを選択します。
- ※ Show In Navigator - **Basic > Navigator** ビューを開き (開かれていない場合)、ファイルシステムツリーを展開して該当するリソースを表示します。
- ※ Copy - 問題情報をシステムのクリップボードにコピーします。
- ※ Paste - システムクリップボードにある問題情報 (該当する場合) を指定されたテキストエディターのカーソルの場所に貼り付けます。
- ※ Delete - 選択した問題の行を削除します (該当する場合)。
- ※ Select All - テーブルのすべての問題を選択します。
- ※ Quick Fix - (Teiid Designer には実装されていません)。
- ※ Properties - 追加情報が含まれるダイアログを表示します。

## D.13. 検索結果ビュー

以下は検索結果の例になります。ビューには検索パラメーターに一致したものを表す行が含まれます。エントリをダブルクリックするとオブジェクトが開かれ、エディターや Model Explorer で選択されます (該当する場合)。

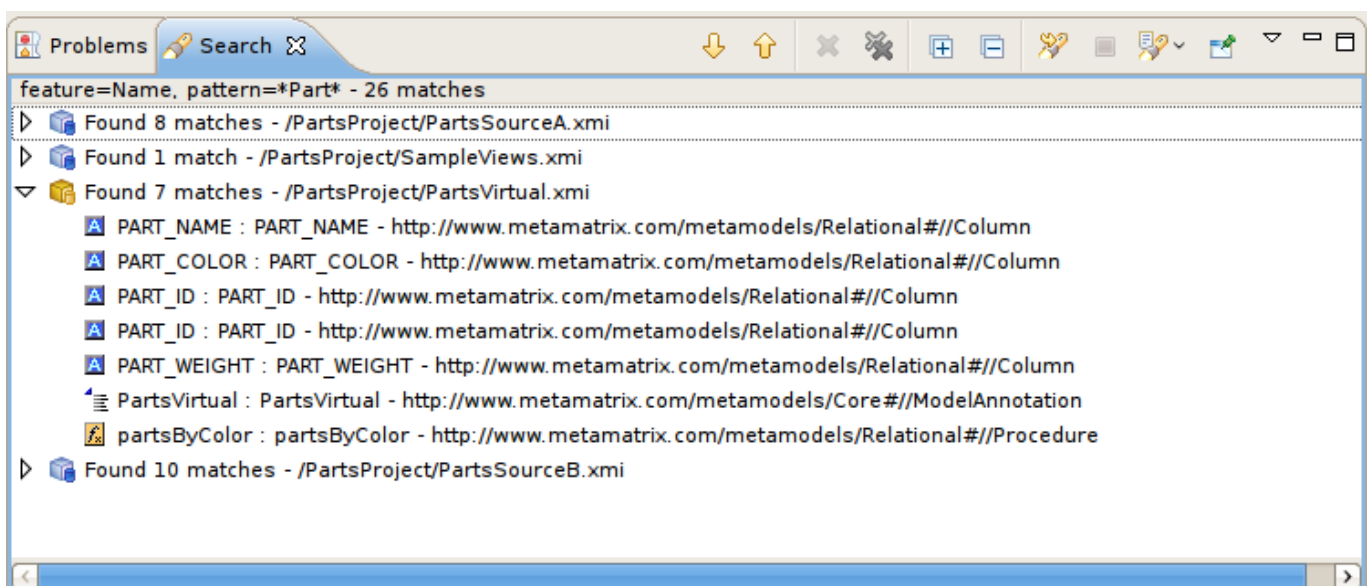


図D.23 検索結果ビュー

**Search Results** ビューのツールバーアクションは次のとおりです。

- ✧ ⬇ Show Next Match - Navigates down one row in the view.
- ✧ ⬆ Show Previous Match - Navigates up one row in the view.
- ✧ ✕ Remove Selected Matches - Removes selected results from the view.
- ✧ ✖ Remove All Matches - Clears the view.
- ✧ 🔍 Search - Launches the Teiid Designer Dialog.
- ✧ 📄 Previous Search Results - Select previous search results from history.

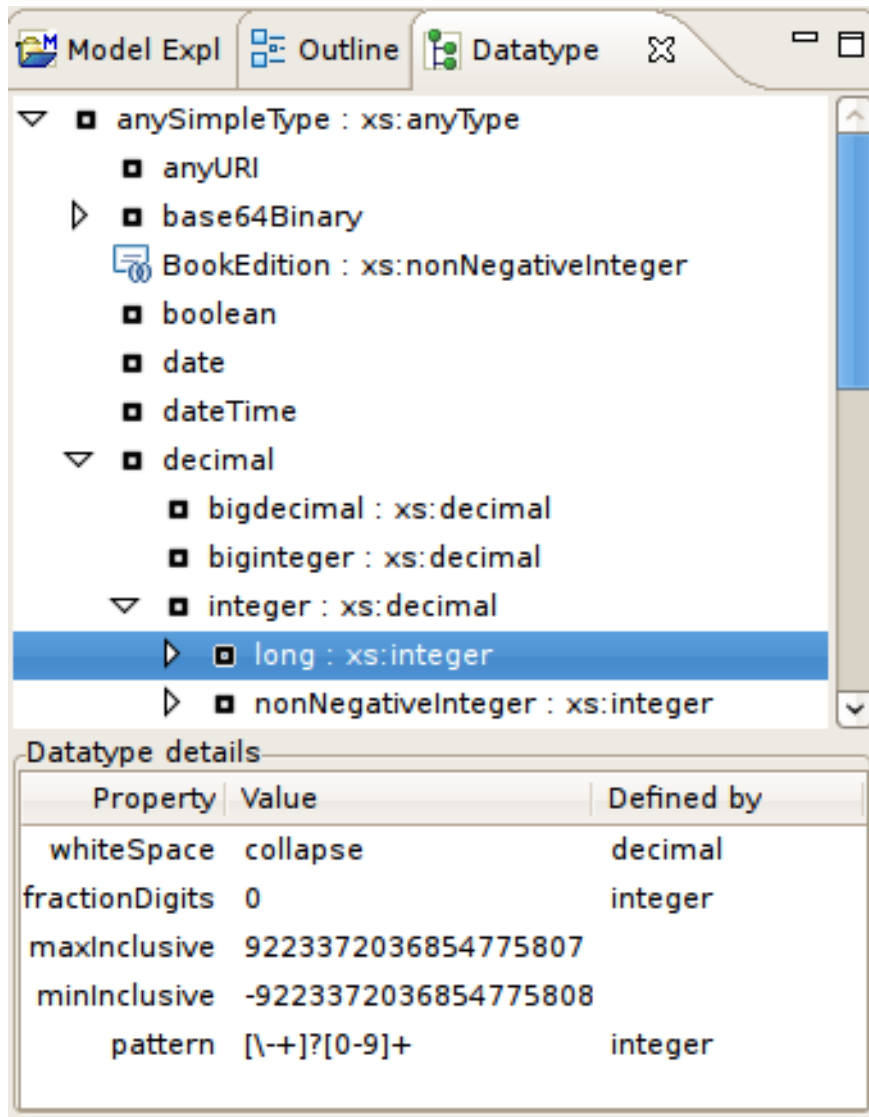
これらのアクションの一部は、右クリックのメニューでも実行できます。



図D.24 検索結果コンテキストメニュー

## D.14. Datatype Hierarchy ビュー

Teiid Designer の **Datatype Hierarchy** ビューを開くには、メインメニューで **Window > Show View > Other...** と選択した後、ダイアログの **Teiid Designer > Datatypes** ビューをクリックします。



図D.25 Datatype Hierarchy ビュー

以下の表は、Teiid Designer タイプと JBoss Data Virtualization Runtime タイプとのマッピングを表しています。

表D.1 対応するランタイムタイプ

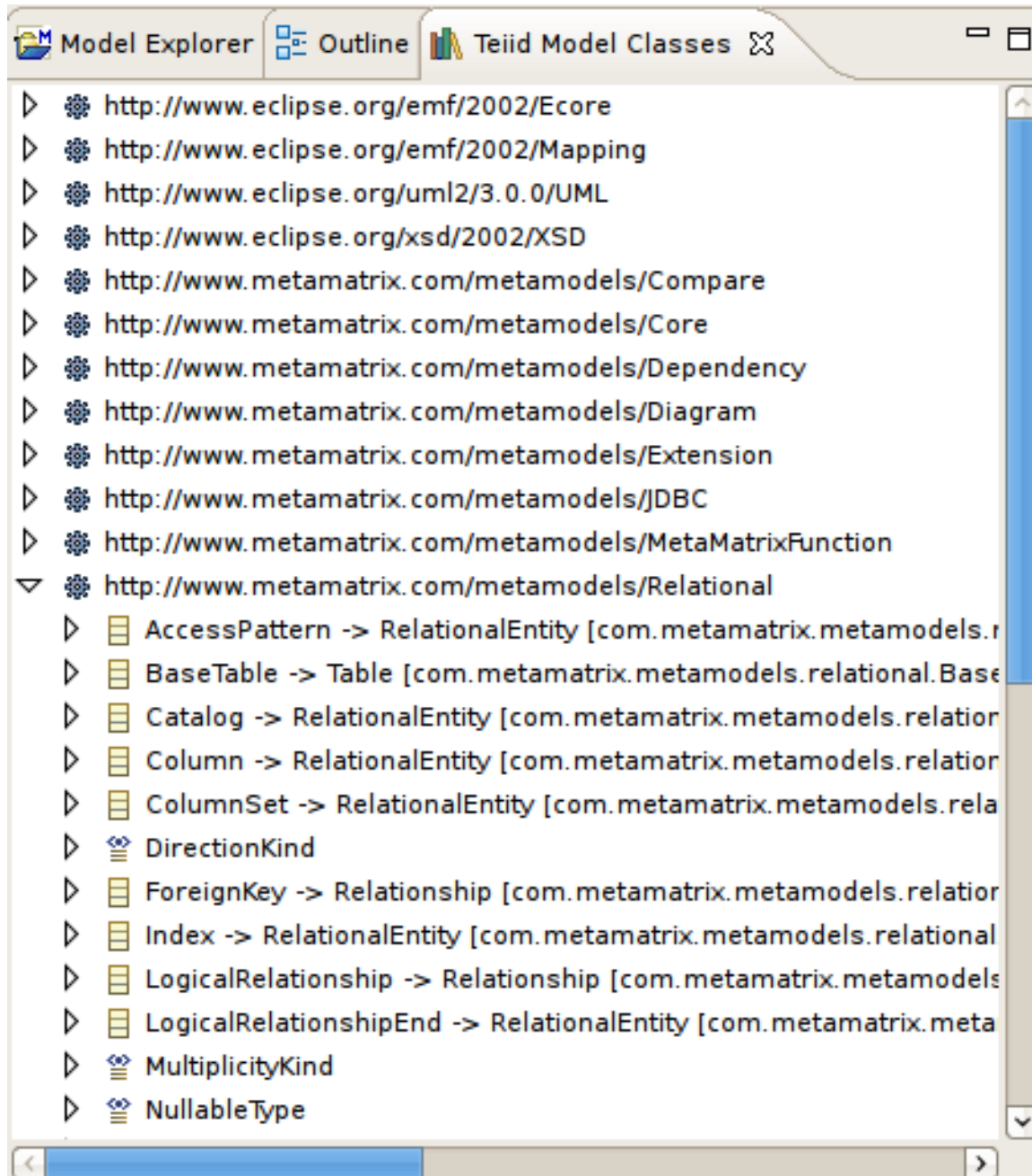
| Teiid Designer タイプ | Java Runtime タイプ     |
|--------------------|----------------------|
| anyURI             | java.lang.String     |
| base64Binary       | java.lang.String     |
| bigdecimal         | java.math.BigDecimal |
| bigint             | java.math.BigInteger |
| blob               | java.sql.Blob [a]    |
| boolean            | java.lang.Boolean    |
| byte               | java.lang.Byte       |
| char               | java.lang.Character  |



| Teiid Designer タイプ  | Java Runtime タイプ     |
|---|----------------------|
| clob  | java.sql.Clob [b]    |
| date  | java.sql.Date        |
| dateTime  | java.sql.Timestamp   |
| decimal   | java.math.BigDecimal |
| double  | java.lang.Double     |
| duration  | java.lang.String     |
| ENTITIES  | java.lang.String     |
| ENTITY  | java.lang.String     |
| float   | java.lang.Float      |
| gDay  | java.math.BigInteger |
| gMonth  | java.math.BigInteger |
| gMonthDay   | java.sql.Timestamp   |
| gYear   | java.math.BigInteger |
| gYearMonth  | java.sql.Timestamp   |
| hexBinary   | java.lang.String     |
| ID  | java.lang.String     |
| IDREF   | java.lang.String     |
| IDREFS  | java.lang.String     |
| int   | java.lang.Integer    |
| 整数  | java.math.BigInteger |
| 言語  | java.lang.String     |
| long  | java.lang.Long       |
| 名前  | java.lang.String     |
| NCName  | java.lang.String     |
| negativeInteger   | java.math.BigInteger |
| NMTOKENS  | java.lang.String     |
| NMTOKENS  | java.lang.String     |
| nonNegativeInteger  | java.math.BigInteger |
| nonPositiveInteger  | java.math.BigInteger |
| normalizedString  | java.lang.String     |
| NOTATION  | java.lang.String     |
| object  | java.lang.Object     |
| positiveInteger   | java.math.BigInteger |
| QName   | java.lang.String     |
| short   | java.lang.Short      |
| 文字列   | java.lang.String     |
| time  | java.sql.Time        |
| timestamp   | java.sql.Timestamp   |
| token   | java.lang.String     |
| unsignedByte  | java.lang.Short      |
| unsignedInt   | java.lang.Long       |
| unsignedLong  | java.lang.BigInteger |
| unsignedShort   | java.lang.Integer    |
| XMLLiteral  | java.sql.SQLXML [c]  |
| <p>[a] 期待される具象タイプは org.teiid.core.types.BlobType です。</p> <p>[b] 期待される具象タイプは org.teiid.core.types.ClobType です。</p> <p>[c] 期待される具象タイプは org.teiid.core.types.XMLType です。</p> |                      |

## D.15. Teiid Model Classes ビュー

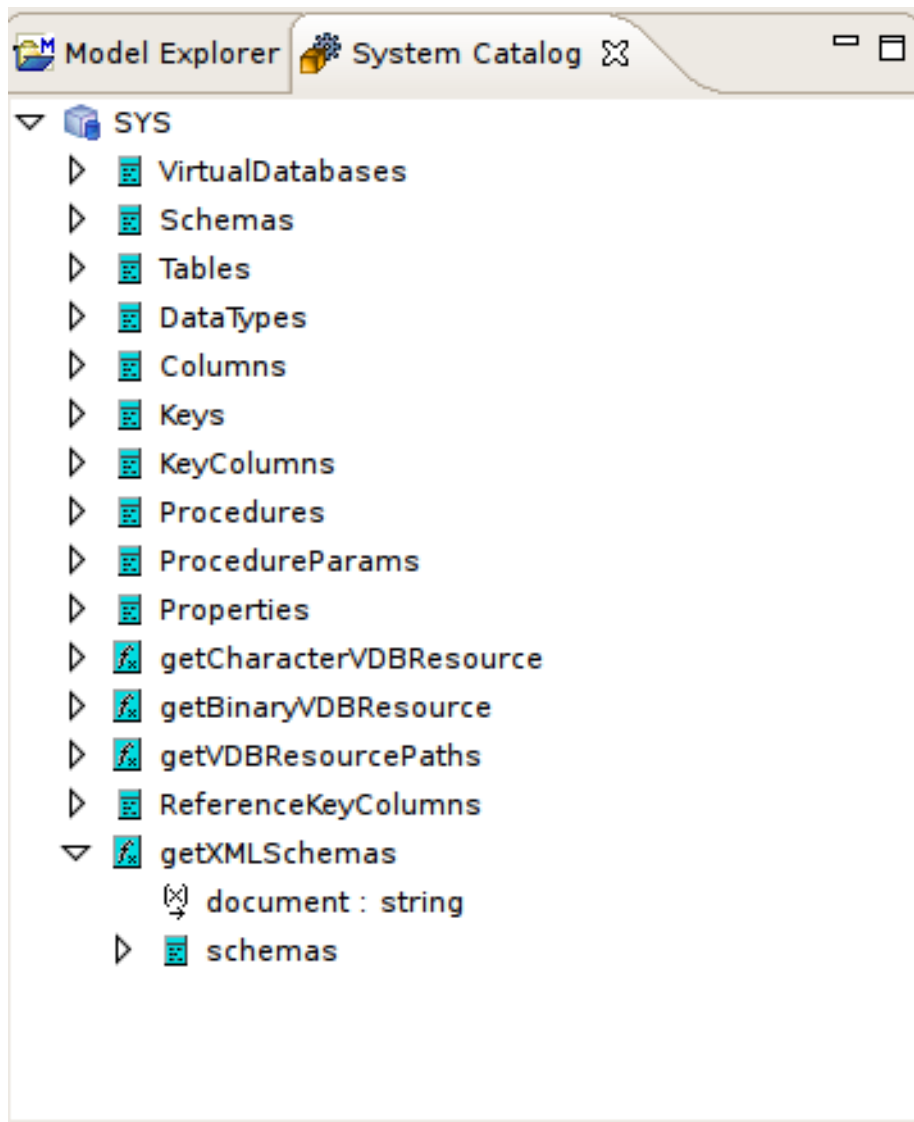
**Model Classes View** は、**Teiid Designer** 内で利用可能なさまざまなメタモデルクラスの階層的で EMF 中心のビューを提供します。このビューの主な目的は情報を提供することですが、特定のメタモデルコンストラクトの関係作成やワークスペース検索で参照として使用することができます。



図D.26 Datatype Hierarchy ビュー

## D.16. Model Extension Registry ビュー

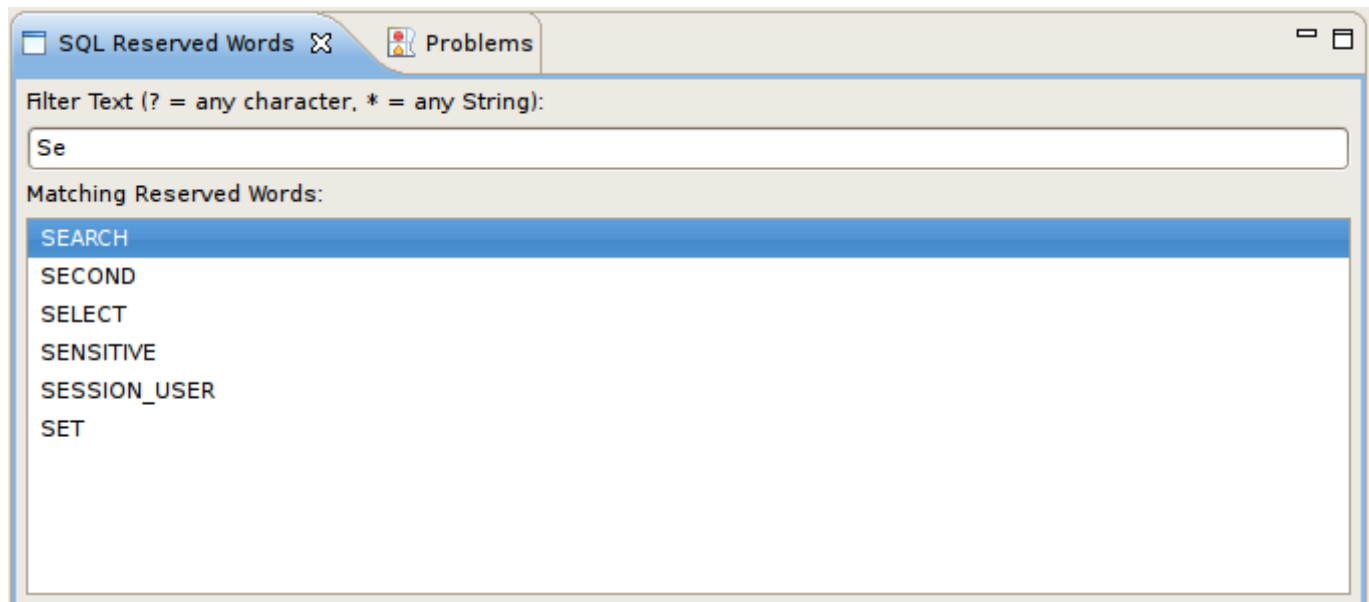
**Teiid Designer** の **System Catalog View** を開くには、メインメニューで **Window > Show View > Other...** とクリックした後、ダイアログの **Teiid Designer > System Catalog** ビューをクリックします。



図D.27 Model Extension Registry ビュー

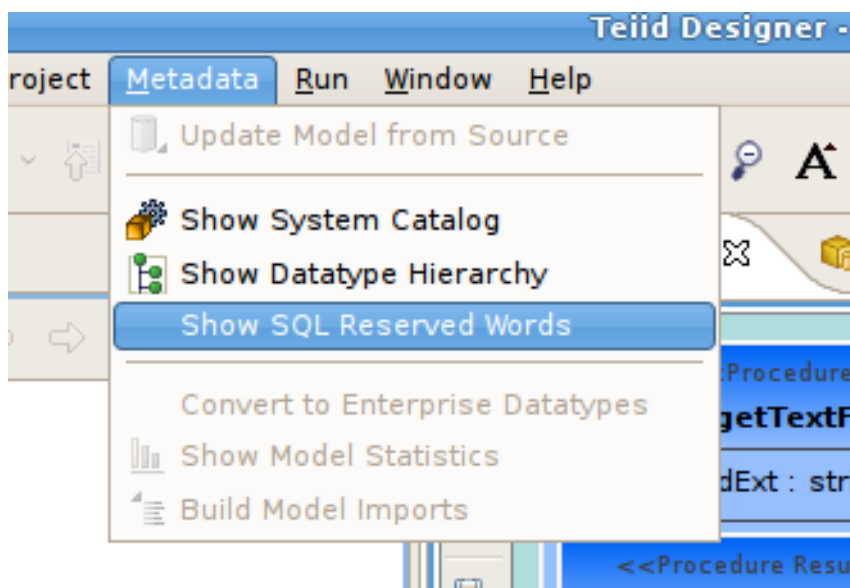
## D.17. SQL Reserved Words ビュー

Teiid Designer の **SQL Reserved Words View** を開くには、メインメニューで **Window > Show View > Other...** とクリックした後、ダイアログの **Teiid Designer > SQL Reserved Words** ビューをクリックします。



図D.28 Model Extension Registry ビュー

また、以下のようにメインメニューで **Metadata > Show SQL Reserved Words** と選択してもビューを表示することができます。



図D.29 SQL Reserved Words アクション

## D.18. Model Extension Definition Registry ビュー (MED Registry ビュー)

Teiid Designer の **MED Registry View** を開くには、メインメニューで **Window > Show View > Other...** とクリックした後、ダイアログの **Teiid Designer > Model Extension Registry** ビューをクリックします。

Model Extension Registry ビューには、現在登録されている MED が表示されます。登録された MED は、ワークスペースのモデルに適用することができます (モデルオブジェクト拡張の管理を参照)。Model Extension Registry ビューは以下のようになります。

| Built-In                            | Namespace Prefix | Namespace URI                               | Model Class | Version | Description                    |
|-------------------------------------|------------------|---|-------------|---------|--------------------------------|
| <input checked="" type="checkbox"/> | ext-custom       | org.teiid.designer.extension.deprecated     | Relational  | 1       | Designer 7.4 extension proper  |
|                                     | mymodelextension | mymodelextension                            | Relational  | 1       | This is my model extension     |
| <input checked="" type="checkbox"/> | rest             | org.teiid.designer.extension.rest           | Relational  | 1       | REST extended virtual procedu  |
| <input checked="" type="checkbox"/> | salesforce       | org.teiid.designer.extension.salesforce     | Relational  | 1       | Salesforce extended relational |
| <input checked="" type="checkbox"/> | sourcefunction   | org.teiid.designer.extension.sourcefunction | Relational  | 1       | Source Function properties     |

図D.30 Model Extension Registry ビュー

共有のサブエディターヘッダーセクションの右隅にある **MED Editor** ツールバーアクションをクリックしてもビューを開くことができます。

登録されている各 MED に対して、名前空間の接頭辞、名前空間の URI、拡張モデルクラス、バージョン、および説明が表示されます。さらに、MED が組み込みであるかどうかを示すフラグが表示されます。Model Extension Registry ビューには、ワークスペース MED ファイルを登録するアクション、ユーザー定義の MED を登録解除するアクション、および登録されている MED とワークスペースにコピーするアクションが含まれます。これらのアクションはすべてコンテキストメニューからでも使用できます。

MED レジストリーは、ワークスペースに登録されたすべての MED をトラッキングします。モデルの拡張には登録された MED のみを使用できます。レジストリーに保存される MED には 2 つの種類があります。

- ※ ビルトイン MED: これらの MED は Teiid Designer のインストール中に登録されます。ユーザーは更新や登録解除することはできません。
- ※ ユーザー定義 MED: これらの MED はユーザーによって作成されます。ユーザーは更新、登録、および登録解除することができます。

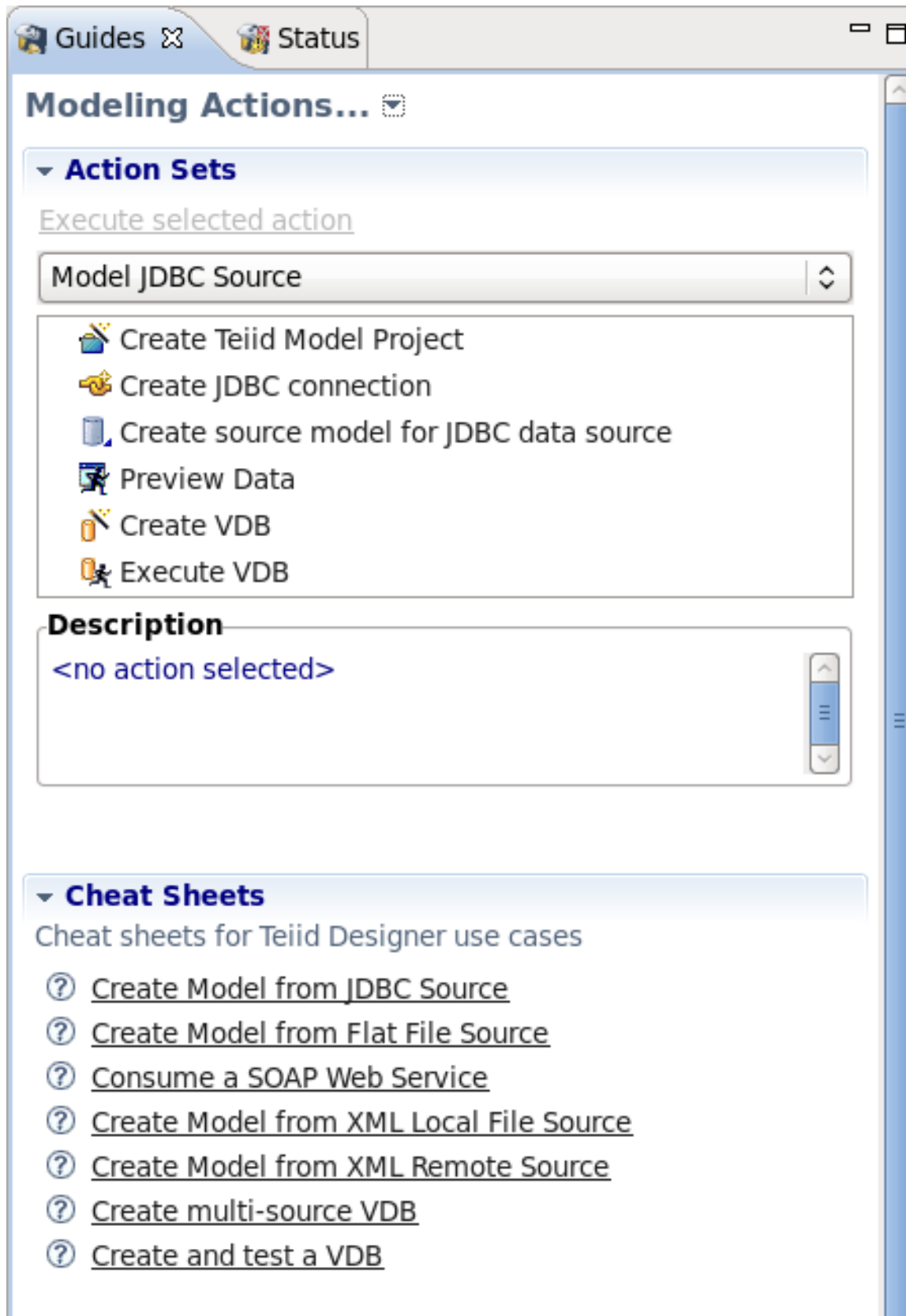
### 注記

ワークスペース MED が登録されたら、必要に応じてワークスペースから削除できます。レジストリーは独自のコピーを保持します。また、適切なツールバーまたはコンテキストメニューアクションを使用すれば、登録されている MED をワークスペースにコピーすることができます。

## D.19. Guides ビュー

Teiid Designer の **Guides View** を開くには、メインメニューで **Window > Show View > Other...** とクリックした後、ダイアログの **Teiid Designer > Guides** ビューをクリックします。

**Guides** ビューは、多くの一般的なモデリング作業を容易にします。このビューにはカテゴリに分類されたモデリングアクションが含まれ、一般的な処理のチートシートへのリンクも含まれています。チートシートは Eclipse の概念で、**Teiid Designer** が貢献しています (Cheat Sheets ビューを参照)。Guides ビューを以下に示します。



図D.31 Guides ビュー

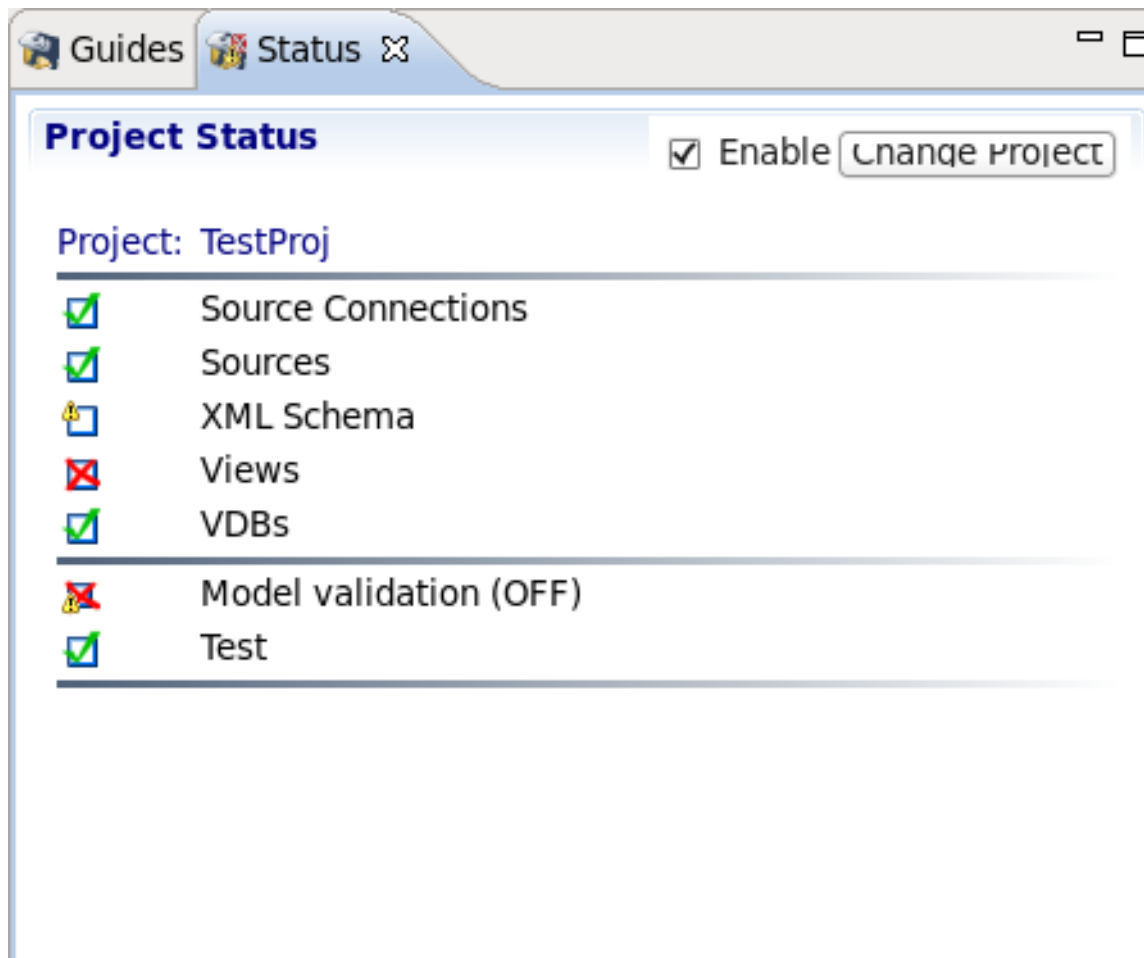
上部の **Action Sets** セクションには、カテゴリに分類されたアクションセットが表示されます。ドロップダウンメニューでカテゴリを選択すると、選択したカテゴリの関連アクションのリストが下に表示されます。**Execute selected action** リンクをクリックするか、アクションをダブルクリックして実行します。

下部の **Cheat Sheets** セクションには、利用可能なチートシートのリンクがリストで表示されます。チートシートが起動されると、選択した処理がステップごとに説明されます。

## D.20. Status ビュー

Teiid Designer の **Status View** を開くには、メインメニューで **Window > Show View > Other...** とクリックした後、ダイアログの **Teiid Designer > Status** ビューをクリックします。

**Status View** は、選択したプロジェクトの概要を表示します。以下にプロジェクトの Status ビューの例を示します。



図D.32 Status ビュー

Status ビューは一般的なプロジェクトエリアに分割されます。

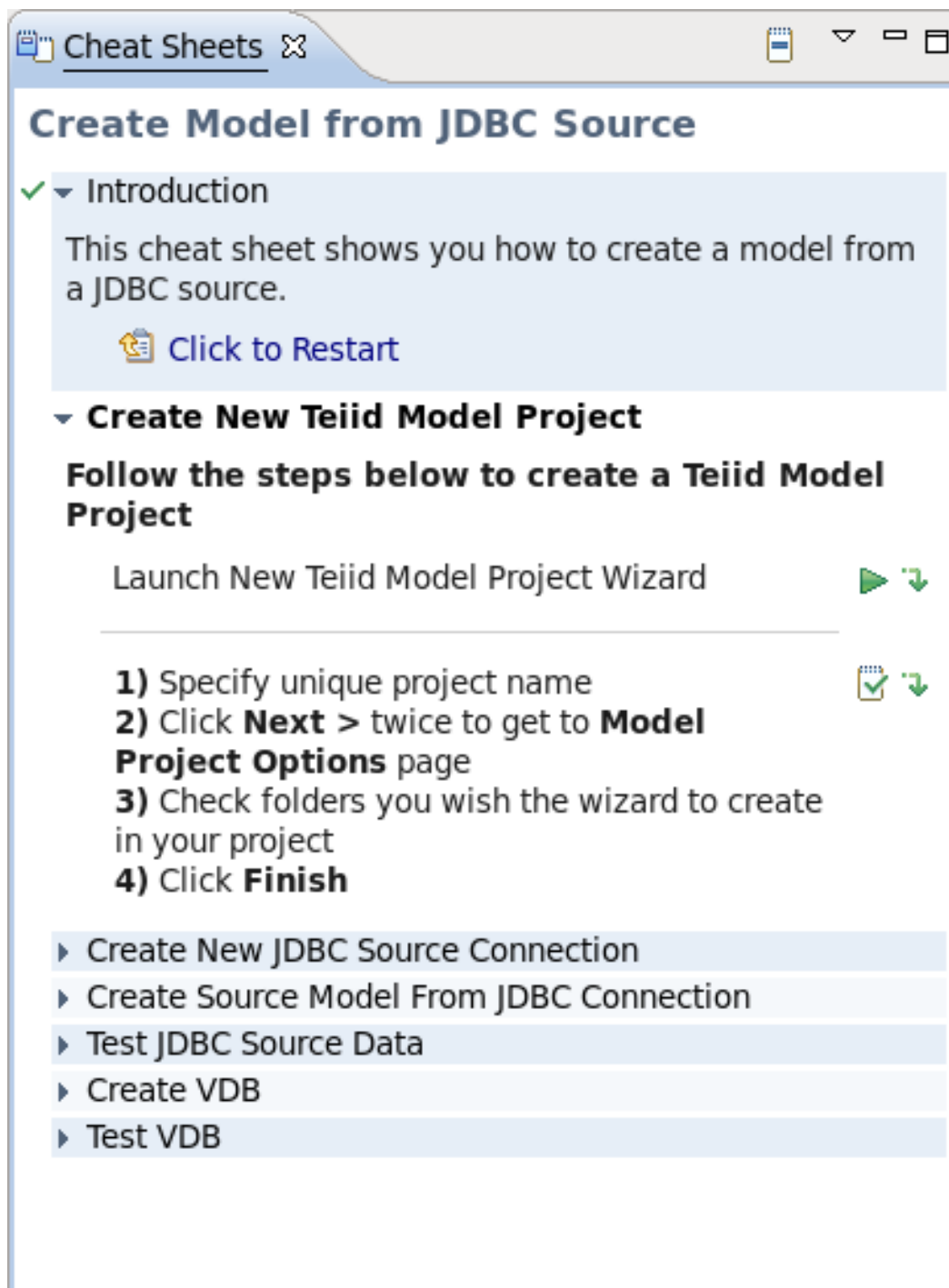
- ※ Source Connections - すべてのソース接続が完全に定義されているか。
- ※ Sources - ソースモデルが存在するか。
- ※ XML Schema - XML スキーマが存在するか。
- ※ Views - ビューモデルが存在するか。
- ※ VDBs - VDB が存在し、デプロイ可能であるか。
- ※ Model Validation (Status) - すべてのモデルが検証に合格したか。
- ※ Test - 定義された VDB がすべて検証に合格したか。

各エリアの状態はアイコンで示されます。緑色のチェックマークは OK を意味します。赤い x はエラーおよび警告のアイコンで潜在的な問題があることを示します。 **Change Project** ボタンを選択するとプロジェクトを変更できます。

## D.21. Cheat Sheets ビュー

**Cheat Sheets View** を開くには、メインメニューで **Window > Show View > Other...** とクリックした後、ダイアログの **Help > Cheat Sheets** ビューをクリックします。

Cheat Sheets ビューは標準の Eclipse Help の概念です。チートシートは、一般的なプロセスワークフローをステップごとに説明します。 **Teiid Designer** は Eclipse Help フレームワークに貢献し、多くの一般的なモデリング作業に関して協力しました。前述のとおり、Guides ビュー (Guides ビューを参照) はこれらのチートシートへのリンクを提供します。チートシートの例を以下に示します。



図D.33 チートシートの例



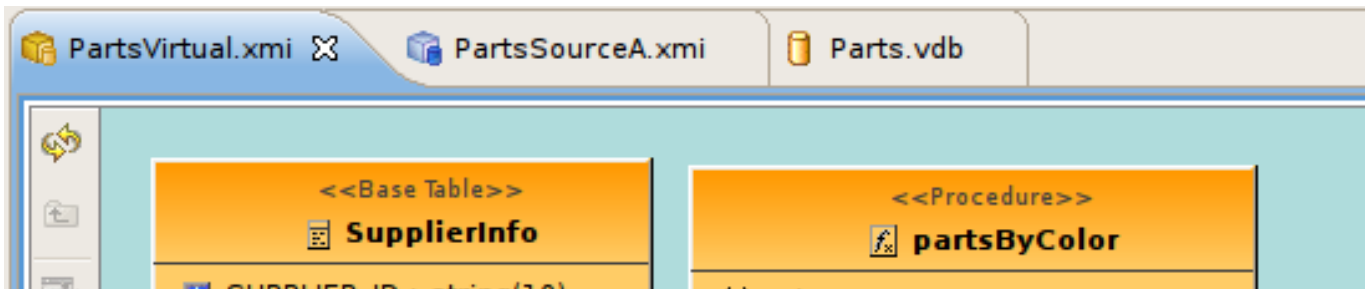
## 付録E エディター

### E.1. エディター

エディターは、モデルの編集を容易にし、ワークスペースのモデルやリソースの状態を維持することが目的のUIコンポーネントです。モデルの編集時、モデルは Model Editor で開かれます。たとえば、プロパティ値の編集には、プロパティを実際に変更する前にエディターを開く必要があります。

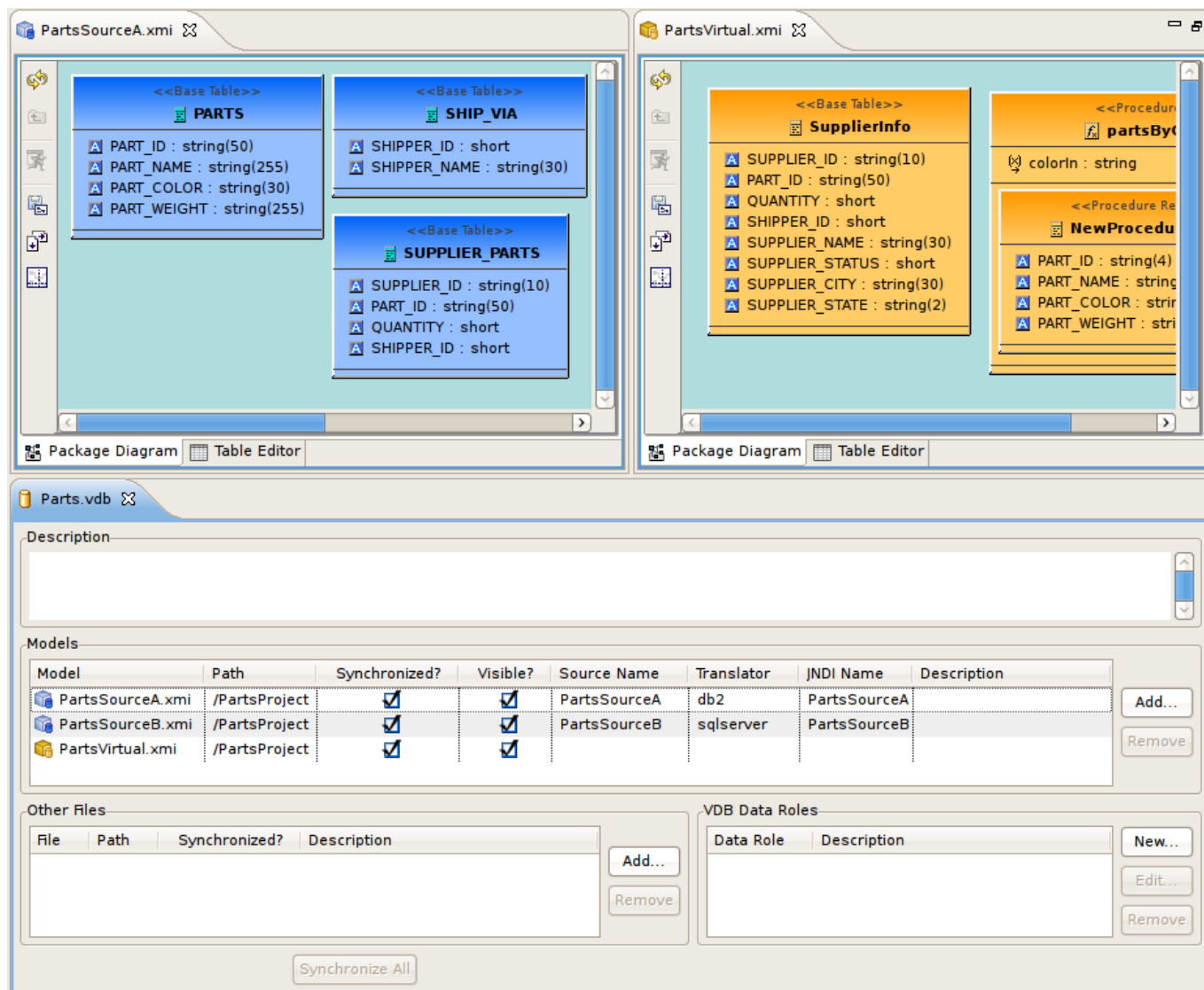
同時に複数のエディターを開くことができますが、1つエディターのみがアクティブになります。Teiid Designer のメインメニューやツールバーにはアクティブなエディターに適用できる演算 (エディターが非アクティブになったら削除できる) が含まれていることがあります。

エディターエリアのタブには、編集するために現在開かれているモデルの名前が表示されます。アスタリスク (\*) はエディターに保存されていない変更があることを示します。



図E.1 エディタータブ

デフォルトでは、エディターはエディターエリアに重ねて表示されますが、複数のモデルを同時に表示するために縦または横に並べることもできます。



図E.2 複数エディターの表示

**Teiid Designer** は、XMI モデルおよび VDB のメインエディタービューを表示します。

**Model Editor** には、XMI モデル内のデータまたはデータの一部の異なるビューを提供するサブエディターが含まれます。モデルタイプ固有のサブエディターは以下のとおりです。

- ✦ Diagram Editor - XML スキーマモデル以外のすべてのモデル。
- ✦ Table Editor - すべてのモデル。
- ✦ Simple Datatypes Editor - XML スキーマモデルのみ。
- ✦ Semantics Editor - XML スキーマモデルのみ。
- ✦ Source Editor - XML スキーマのみ。

**VDB Editor** は、説明、モデルのコンテンツ、およびデータのルールを編集するパネルが含まれる単一ページのエディターです。

通常のモデルエディター以外に、特定のモデルオブジェクトタイプを編集するための詳細なエディターがあります。これらのオブジェクトエディターには以下が含まれます。

- ✦ Transformation Editor - リレーショナルビューベーステーブル、プロシージャ、および XML Web サービス演算のトランスフォーメーション SQL を管理します。

- ※ Choice Editor - XML ドキュメントビューモデルの XML choice 要素のプロパティおよび基準を管理します。
- ※ Input Editor - XML ドキュメントビューモデルのマッピングクラス間で使用される入力セットパラメータを管理します。
- ※ Recursion Editor - XML ドキュメントビューモデルの再帰 XML 要素に対する再帰プロパティを管理します。
- ※ Operation Editor - Web サービス演算の SQL および入力変数を管理します。

## E.2. Model Editor






### E.2.1. Model Editor

**Model Editor** は、データの複数のビューを提供するサブエディターで構成されます。**Diagram Editor** はグラフィカルな編集機能を提供し、**Table Editor** はスプレッドシートのような編集機能を提供します。ここではさまざまなサブエディターについて説明します。

### E.2.2. Diagram Editor

**Diagram Editor** はモデルコンポーネントのセットとそれらの関係のグラフィカルなビューを表示します。

モデルタイプに応じて複数のタイプのダイアグラムを使用できます。以下のダイアグラムが含まれます。

- ※  Package Diagram
- ※  Custom Diagram
- ※  Transformation Diagram
- ※  Mapping Diagram
- ※  Mapping Transformation Diagram

ダイアグラム設定でさまざまなダイアグラムビジュアルプロパティをカスタマイズできます。

各ダイアグラムは、メインツールバー、ダイアグラムツールバー、および選択ベースのコンテキストメニューよりアクションを提供します。ダイアグラムタイプごとのこれらのアクションの詳細は次のとおりです。

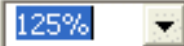




Diagram Editor がフォーカスされると、共通のダイアグラムアクションのセットがアプリケーションのメインツールバーに追加されます。



図E.3 メインツールバーダイアグラムアクション

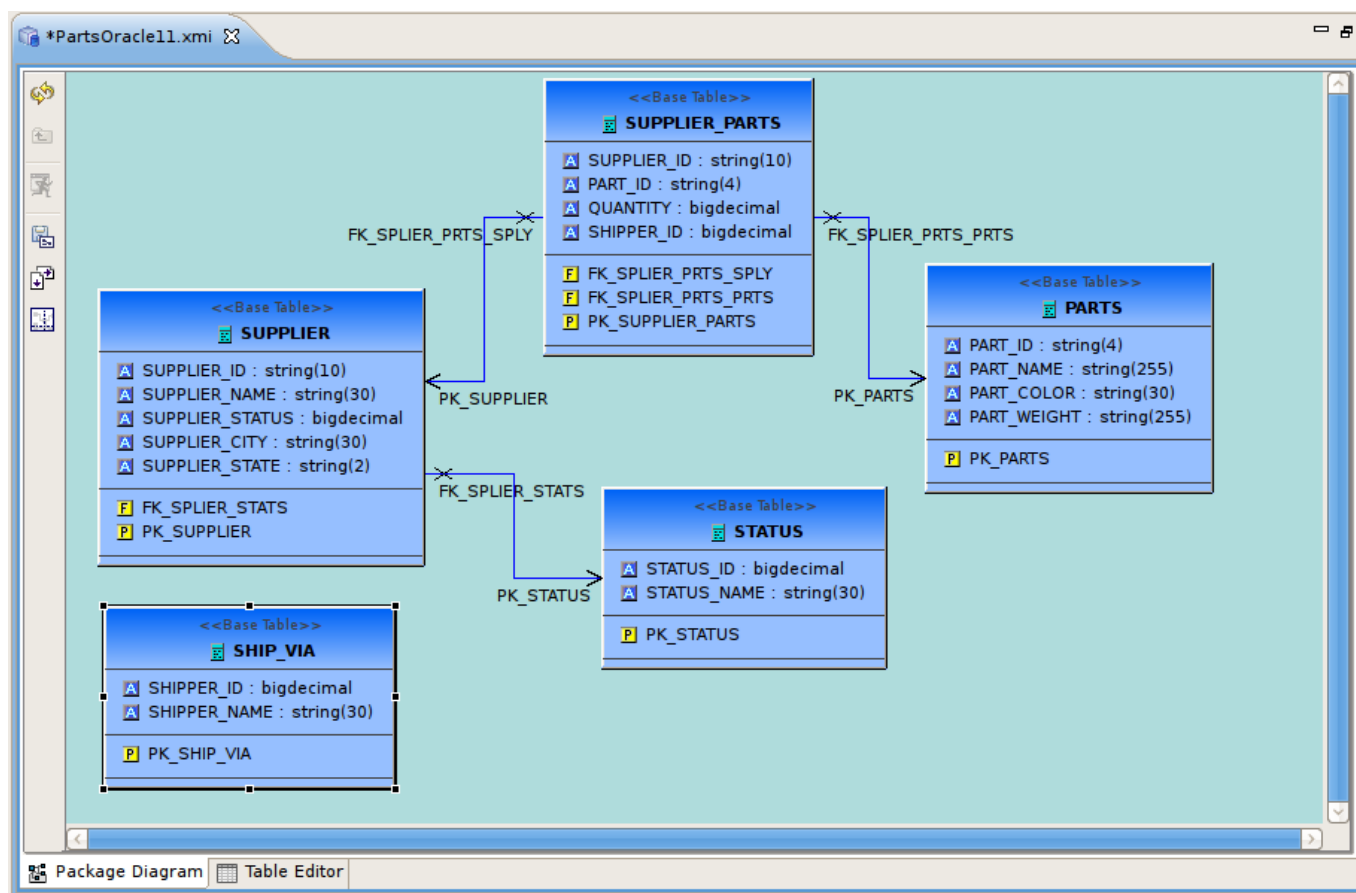
アクションには以下が含まれます。

- ※  Zoom In

- ※  Zoom to Level
- ※  Zoom Out
- ※  Increase Font Size
- ※  Decrease Font Size
- ※  Perform Diagram Layout





### E.2.3. Package Diagram



**Package Diagram** は、モデル自体、リレーショナルカタログ、またはスキーマなど、モデルコンテナのコンテンツのグラフィカルなビューを提供します。



図E.4 Package Diagram の例

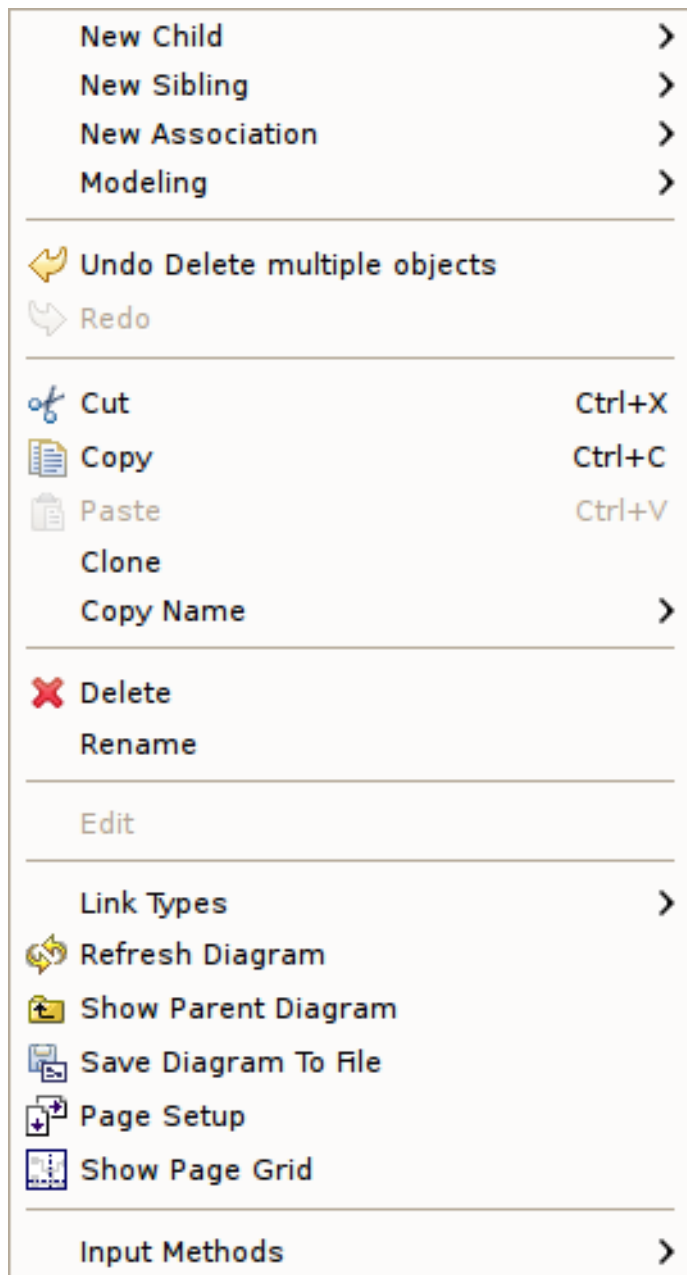
Package Diagram ツールバーアクションには以下が含まれます。

- ※  Refresh Diagram - Redraws diagram.
- ※  Show Parent Diagram - Navigates to diagram for parent object (if available).
- ※  Preview Data - Executes a simple preview query (**SELECT \* FROM** ).
- ※  Save Diagram as Image - Save the diagram image to file in JPG or BMP format.

- ✦  Modify Diagram Printing Preferences - Modify page layout information for printing diagrams. Includes margins, orientation, etc...
- ✦  Show/Hide Page Grid - Show current page boundaries as grid in diagram.

コンテキストメニューは、特に Package Diagram からモデルデータを編集する柔軟な方法を提供します。各 Package Diagram はコンテナの内容 (モデル、カテゴリー、スキーマなど) を表すため、標準の Edit アクション (Delete、Cut、Copy、Paste、Rename、Clone) の他にも New Child、New Sibling、および New Association アクションがほぼ常に利用できます。

以下は、リレーショナルベーステーブルのコンテキストメニューの例になります。



図E.5 Package Diagram コンテキストメニュー










### E.2.4. Custom Diagram

**Custom Diagram** はユーザー定義のモデルオブジェクトのビューを表します。Package Diagram とは異なり、Custom Diagram には未関連だけでなく異なるコンテナやイベントからのオブジェクトを含めることができます。



## 図E.6 Custom Diagram の例

Custom Diagram ツールバーアクションには以下が含まれます。

- ※  Refresh Diagram - Redraws diagram.
- ※  Show Parent Diagram - Navigates to diagram for parent object (if available).
- ※  Preview Data - Executes a simple preview query (**SELECT \* FROM** ).
- ※  Add To Diagram - Add objects selected in Model Explorer to diagram.
- ※  Remove From Diagram - Removed objects selected in diagram from diagram.
- ※  Clear Diagram - Remove all objects from diagram.
- ※  Save Diagram as Image - Save the diagram image to file in JPG or BMP format.
- ※  Modify Diagram Printing Preferences - Modify page layout information for printing diagrams. Includes margins, orientation, etc...
- ※  Show/Hide Page Grid - Show current page boundaries as grid in diagram.

Custom Diagram はコンテナオブジェクトの内容 (モデル、カテゴリ、スキーマなど) を表さないため、コンテキストメニューはダイアグラムへのオブジェクトの追加、ダイアグラムからのオブジェクトの削除、および基本のダイアグラム関連の表示オプションに限定されます。







## E.2.5. Transformation Diagram






**Transformation Diagram** は、ビューテーブルの SQL トランスフォーメーションに記述されたソース入力によって定義された関係のビューを表します。



## 図E.7 Transformation Diagram の例

Transformation Diagram ツールバーアクションには以下が含まれます。

- ※  Refresh Diagram - Redraws diagram.
- ※  Show Parent Diagram - Navigates to diagram for parent object (if available).
- ※  Preview Data - Executes a simple preview query (**SELECT \* FROM** ).
- ※  Add Transformation Sources - Add selected sources to transformation.
- ※  Add Union Transformation Sources - Add selected sources as union sources.
- ※  Remove Transformation Sources - Removed sources selected in diagram from transformation.

- ※  Clear Transformation - Remove all sources from transformation.
- ※  Open Transformation Reconciler dialog
- ※  Save Diagram as Image - Save the diagram image to file in JPG or BMP format.
- ※  Modify Diagram Printing Preferences - Modify page layout information for printing diagrams. Includes margins, orientation, etc...
- ※  Show/Hide Page Grid - Show current page boundaries as grid in diagram.













## E.2.6. Mapping Diagram

**Mapping Diagram** は仮想マッピングクラスの列と XML ドキュメント要素との間のマッピングのビューを表します。このマッピングは、ソースデータが行ベースの結果から XML 形式のテキストにトランスフォーメーションされる方法を定義します。



図E.8 Mapping Diagram の例

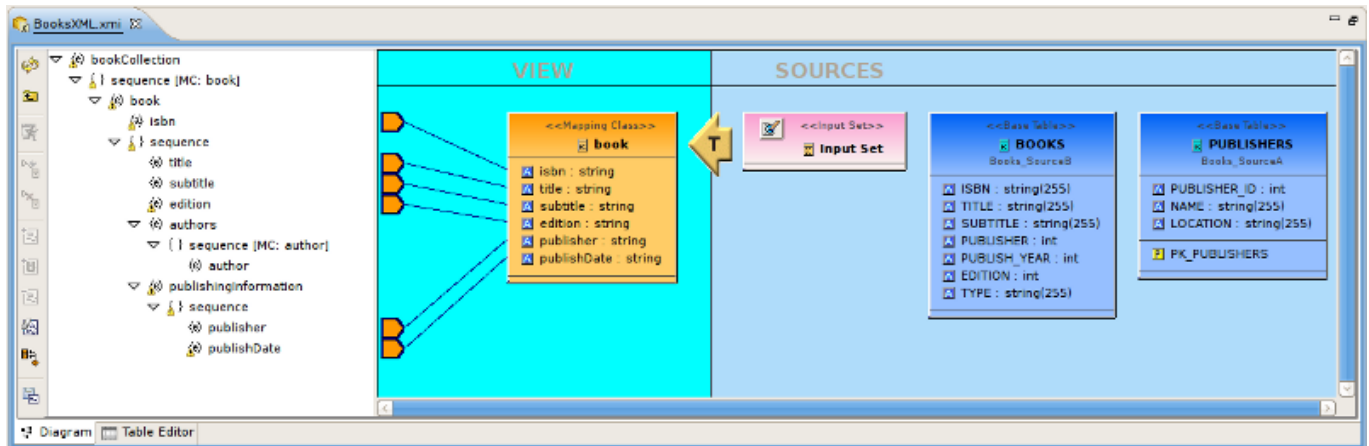
Mapping Diagram ツールバーアクションには以下が含まれます。

- ※  Refresh Diagram - Redraws diagram.
- ※  Show Parent Diagram - Navigates to diagram for parent object (if available).
- ※  Show Mapping Transformation Diagram - Show detailed mapping transformation diagram for selected mapping class.
- ※  Preview Data - Executes a simple preview query (SELECT \* FROM ).
- ※  Generate Mapping Classes - Generate mapping classes for the selected XML document root element.
- ※  New Mapping Class - Insert new mapping class referenced to the selected XML document element or attribute..
- ※  New Staging Table - Insert new staging table referenced to the selected XML document element or attribute.
- ※  Merge Mapping Classes - Merge selected mapping classes.
- ※  Split Mapping Class - Split selected mapping class.
- ※  Display All Mapping Classes
- ※  Show Mapping Class Columns
- ※  Filter Displayed Mapping Classes with Selection

Mapping Diagram コンテキストメニューは、マッピングクラスの操作アクション (Merge Mapping Classes、Split Mapping Class など) の他にマッピングクラスへの編集機能を提供します。

## E.2.7. Mapping Transformation Diagram

Mapping Transformation Diagram は、入力セットやステージングテーブルをマッピングクラスのトランスフォーメーションに対するソースとして表示すること以外は Transformation Diagram と同じです。



図E.9 Mapping Transformation Diagram の例

Mapping Transformation Diagram ツールバーアクションには以下が含まれます。

- ✦ Refresh Diagram - Redraws diagram.
- ✦ Show Parent Diagram - Navigates to diagram for parent object (if available).
- ✦ Preview Data - Executes a simple preview query (**SELECT \* FROM** ).
- ✦ New Mapping Link - Create a mapping link between selected mapping extent (i.e. XML element or attribute) and mapping class column.
- ✦ Remove Mapping Link - Delete mapping link between selected mapping extent (i.e. XML element or attribute) and mapping class column.
- ✦ Add Transformation Sources - Add selected sources to transformation.
- ✦ Add Union Transformation Sources - Add selected sources as union sources.
- ✦ Remove Transformation Sources - Removed sources selected in diagram from transformation.
- ✦ Clear Transformation - Remove all sources from transformation.
- ✦ Open Transformation Reconciler dialog
- ✦ Save Diagram as Image - Save the diagram image to file in JPG or BMP format.
- ✦ Modify Diagram Printing Preferences - Modify page layout information for printing diagrams. Includes margins, orientation, etc...

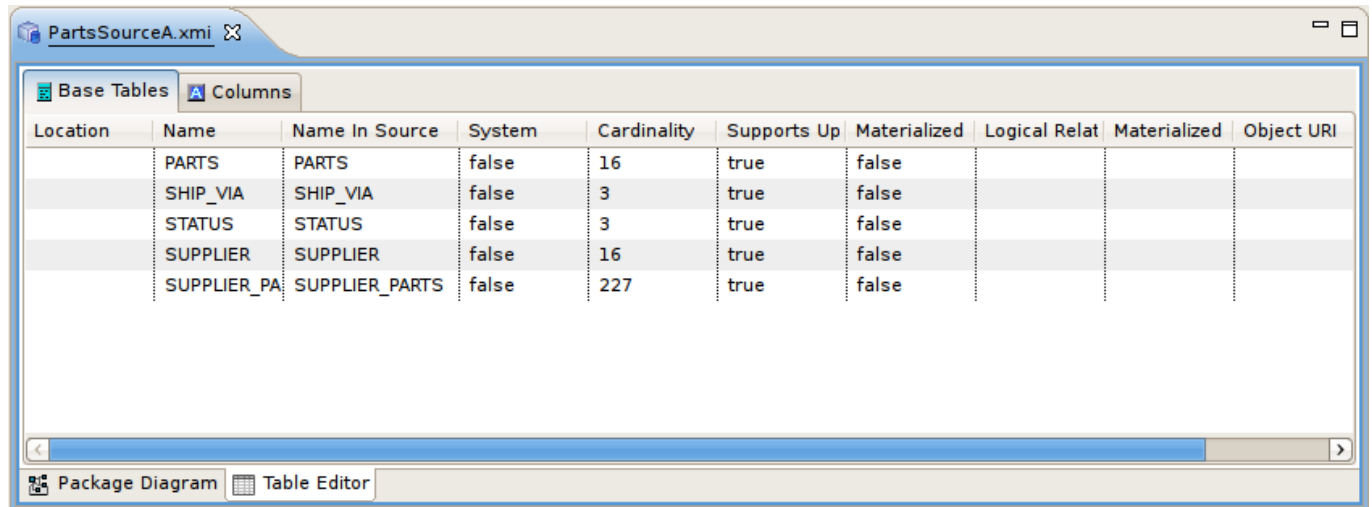
Mapping Transformation Diagram のコンテキストメニューにある機能は Transformation Diagram と同じですが、入力セットの管理および編集が追加されます。

## E.3. Table Editor



### E.3.1. Table Editor

**Table Editor** は、モデルコンテンツのテーブルベースのオブジェクトタイプで構成されたビューを提供します。以下の図は、Table Editor で表示されるリレーショナルモデルを表しています。共通のオブジェクトタイプは個別のフォルダーまたはテーブルに表示されます。たとえば、すべてのベーステーブルは親とは独立した 1 つのテーブルに表示されます。



| Location | Name        | Name In Source | System | Cardinality | Supports Up | Materialized | Logical Relat | Materialized | Object URI |
|----------|-------------|----------------|--------|-------------|-------------|--------------|---------------|--------------|------------|
|          | PARTS       | PARTS          | false  | 16          | true        | false        |               |              |            |
|          | SHIP_VIA    | SHIP_VIA       | false  | 3           | true        | false        |               |              |            |
|          | STATUS      | STATUS         | false  | 3           | true        | false        |               |              |            |
|          | SUPPLIER    | SUPPLIER       | false  | 16          | true        | false        |               |              |            |
|          | SUPPLIER_PA | SUPPLIER_PARTS | false  | 227         | true        | false        |               |              |            |

図E.10 Table Editor の例

**Table Editor Preferences** から **Table Editor** プロパティをカスタマイズできます。

Table Editor の主な機能は次のとおりです。

- ✦ 既存のプロパティの編集。
- ✦ メイン Edit メニューおよびコンテキストメニュー (Cut、Copy、Paste、Clone、Delete、Rename、Insert Rows) からのオブジェクトの追加、削除、または編集。
- ✦ クリップボードの情報をテーブルに貼り付ける。
- ✦ テーブルの出力。




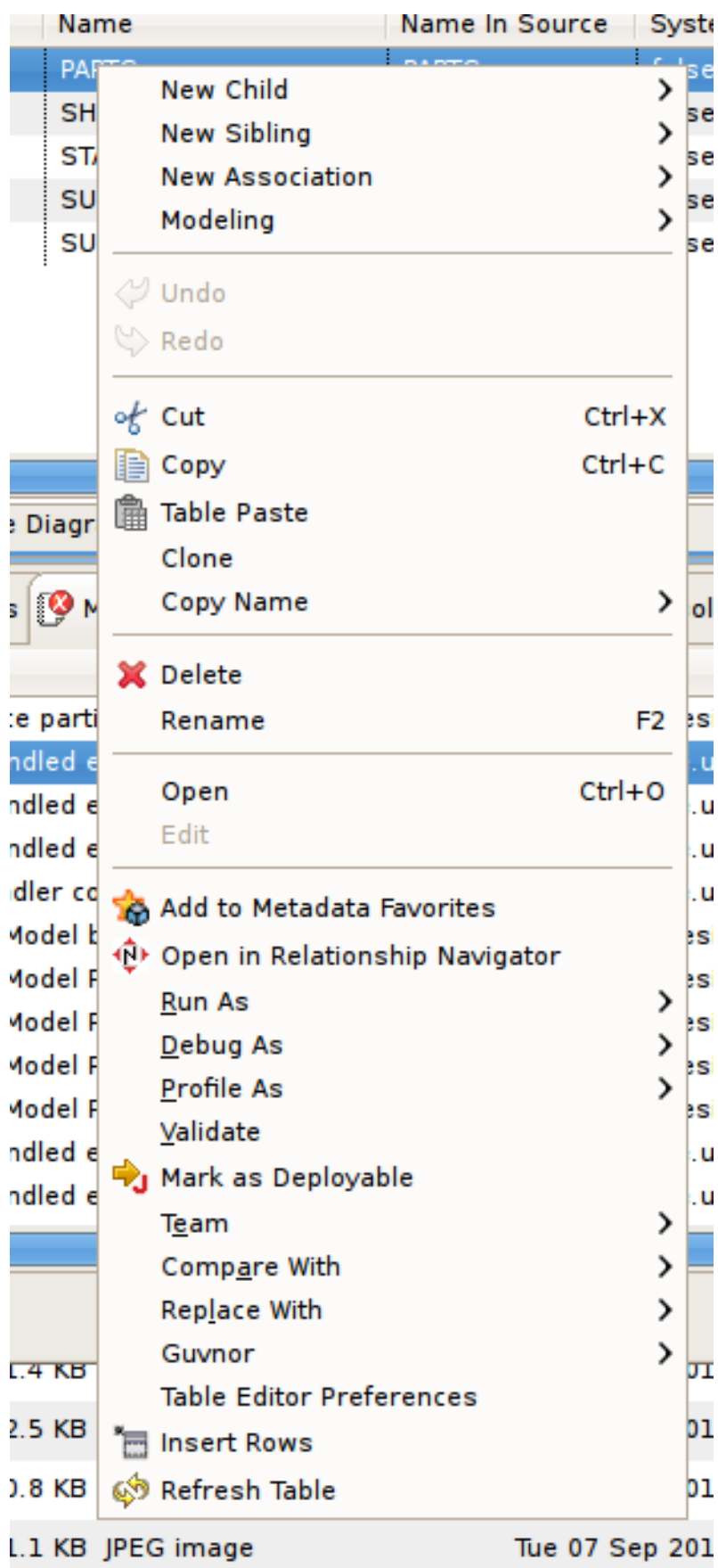
When a Table Editor is in focus, the **Insert Table Rows** action  is added to the application's main toolbar.

Table Editor の一部のアクションは、選択されたテーブル行の右クリックメニューに反映されます。これらのアクションには以下が含まれます。

- ✦  Table Paste - Paste common spreadsheet data (like Microsoft Excel) to set object properties.
- ✦ Table Editor Preferences - 表示されるプロパティのカスタマイズなど、テーブルエディターの設定を変更します。
- ✦ Insert Rows - 新しい兄弟オブジェクトを複数作成します。
- ✦  Refresh Table - Refreshes the contents of the current Table Editor to insure it is in sync with the model.



図E.11 Table Editor の例

### E.3.2. プロパティの編集

テーブルのセルをダブルクリックするとプロパティを編集できます。

String プロパティでは、テーブルのセルがインプレーステキストエディターフィールドになります。

| ation | Name                 | Na |
|-------|----------------------|----|
|       | CATEGORIES           | CA |
|       | CUSTOMERCUSTOMERDEMO | CU |
|       | CUSTOMERDEMOGRAPHICS | CU |
|       | CUSTOMERS            | CU |
|       | EMPLOYEES            | EM |

図E.12 String プロパティの編集

プロパティがブール値 (true または false) タイプであるか、複数の選択可能な値がある場合、値の変更にコンボボックスが表示されます。

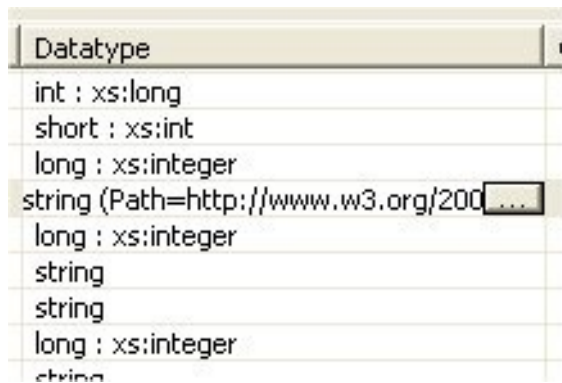
| Supports U... |
|---------------|
| true          |
| true          |
| true          |
| true          |
| true          |
| true          |
| true          |
| true          |
| true          |

図E.13 ブール値の編集

| Searchability   | Current |
|-----------------|---------|
| ALL_EXCEP...    | false   |
| ALL_EXCEP...    | false   |
| ALL_EXCEP...    | false   |
| ARCHABLE        | false   |
| SEARCHABLE      |         |
| ALL_EXCEPT_LIKE |         |
| LIKE_ONLY       |         |
| UNSEARCHABLE    |         |
| SEARCHABLE      | false   |
| SEARCHABLE      | false   |

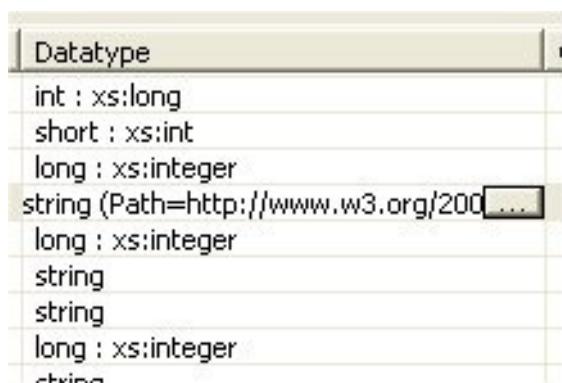
図E.14 複数值プロパティの編集

利用可能な値が動的である (例: 利用可能なモデルやデータを基に変更できる) 複数值プロパティの場合、ピッカーボタン . . . . が表示されます。



図E.15 ピッカーを使った複数値の編集

リレーショナル列データタイププロパティがこのタイプの例になります。テーブルセルで編集し、データタイプの ... ボタンをクリックすると、以下のダイアログが表示されます。



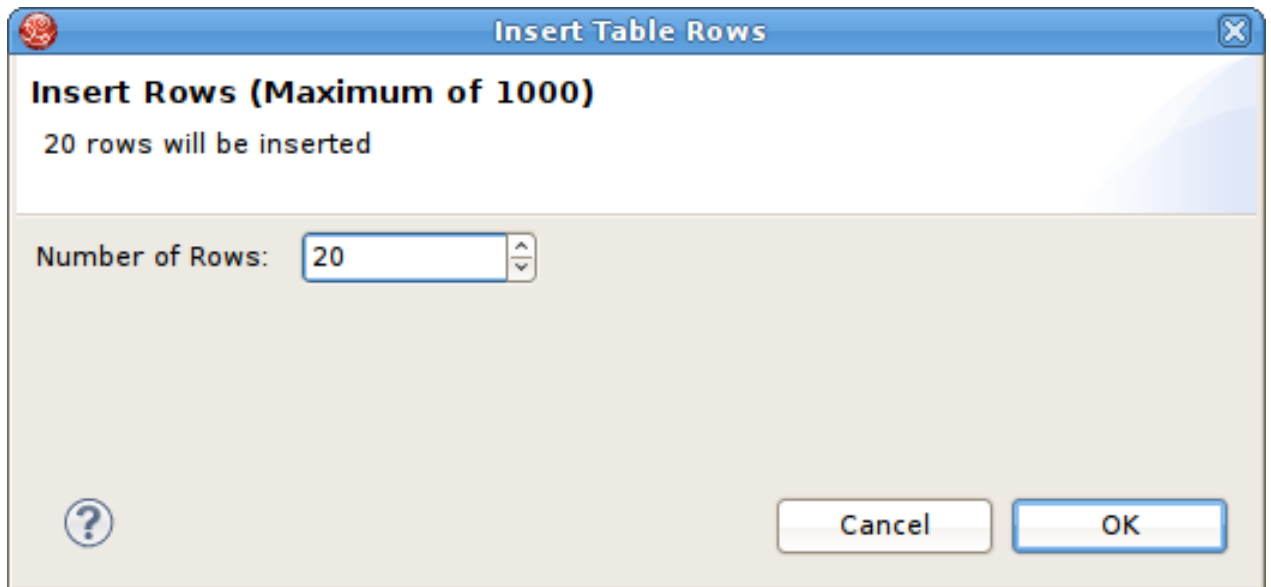
図E.16 データタイプ値の編集

### E.3.3. テーブル行の挿入

**Insert Rows** アクションはモデルでオブジェクトを作成する別の方法を提供します。**Insert Rows** アクションは **Insert Sibling** と同じ機能を実行しますが、同時に複数の子を作成できます。新しい行はすべて選択されたオブジェクトと同じタイプのオブジェクトに対応し、選択されたオブジェクトと同じ親の下に位置します。

テーブルで行を挿入するには、以下を行います。

1. 挿入する行の上になるテーブル行を選択します。
2. 右クリックで **Insert Rows** アクションを選択するか、メインツールバーの **Insert Rows** をクリックします。以下のダイアログが表示されます。



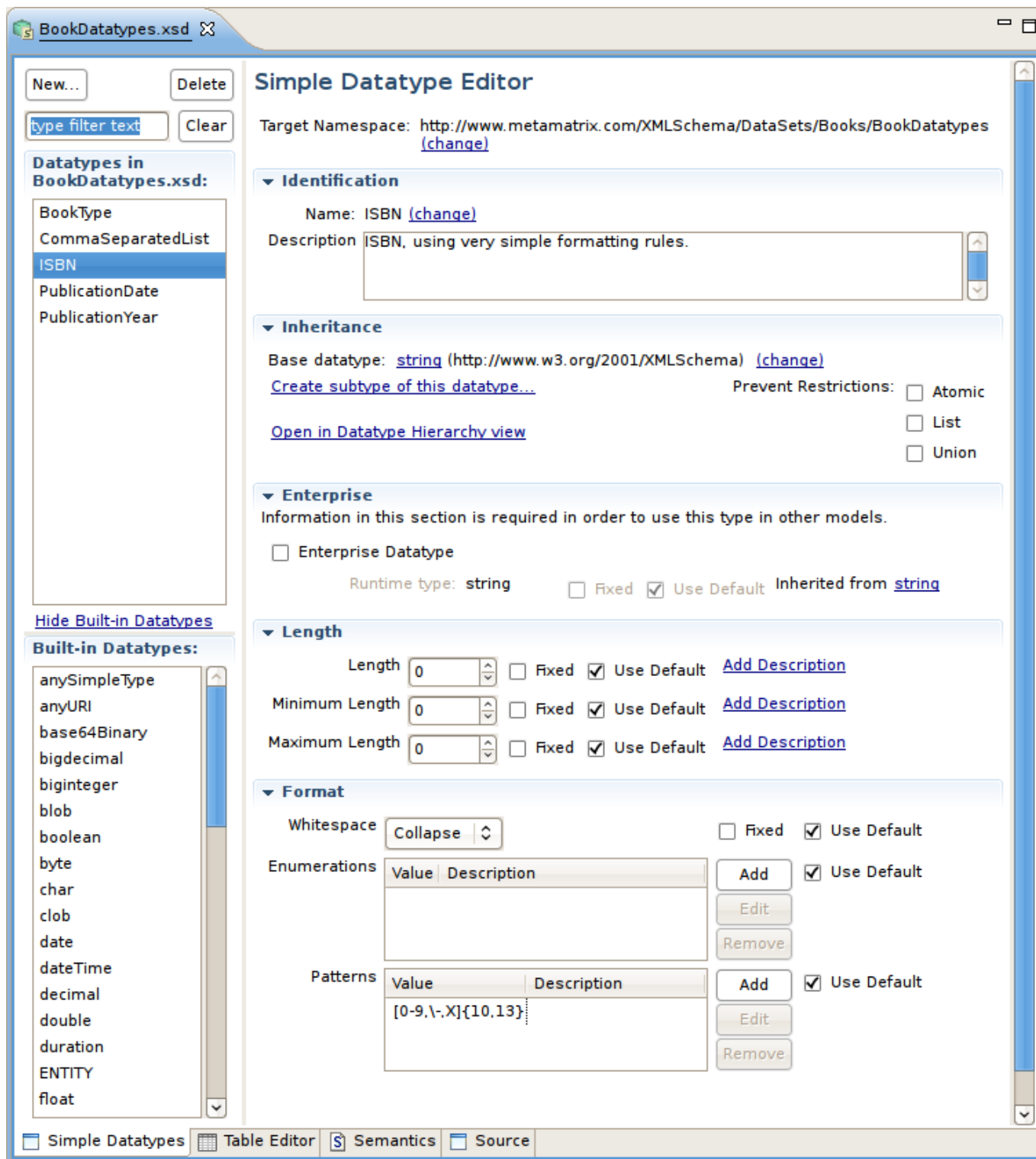
図E.17 String プロパティの編集

3. ダイアログの Number of Rows の値を編集するか、アップダウンボランを使用して値を変更します。
4. ダイアログで **OK** をクリックします。

選択したテーブル行の下に指定した数の行 (新しいモデルオブジェクト) が追加されます。

## E.4. Simple Datatypes Editor

**Simple Datatypes Editor** は XML スキーマデータのフォームベースのプロパティビューを提供します。



図E.18 String プロパティの編集

## E.5. Semantic Editor

**Semantic Editor** は、XML スキーマ要素および属性のツリーベースのエディターです。

## E.6. Source Editor

**Source Editor** は XML スキーマのフォーマットルールを認識する簡単なテキストエディターです。

## E.7. Model Object Editors

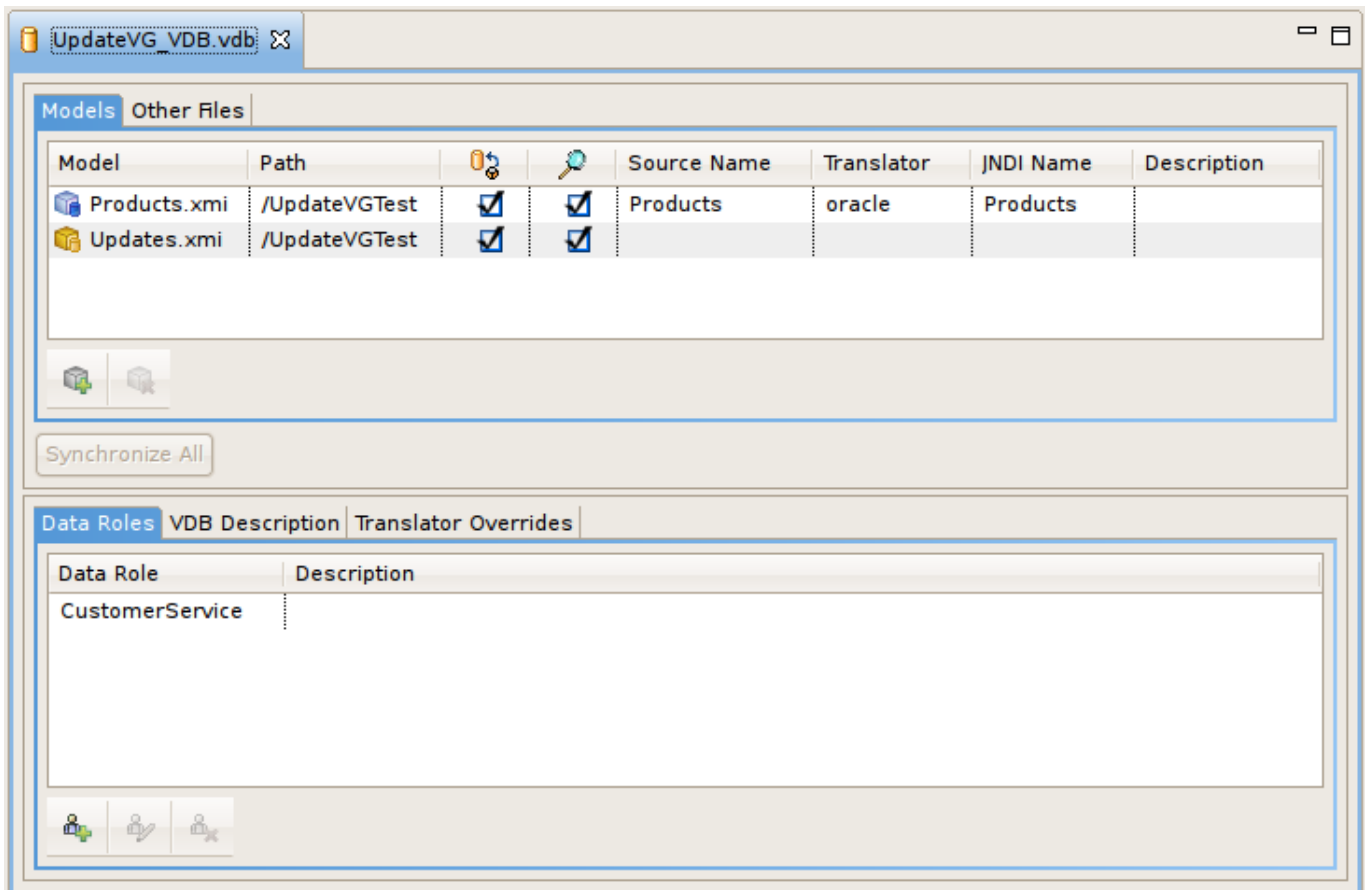
**Model Object Editors** は、特定のモデルオブジェクトタイプに対して使用できる特別なサブエディターを表します。

## E.8. VDB Editor

### E.8.1. VDB Editor

VDB (仮想データベース) は複数のデータソースからデータを統合するために使用されるコンポーネントのコンテナであるため、単一の統一 API を介してフェデレーションにてアクセスできます。VDB にはデータソースの構造的な性質を定義するモデル、ビュー、および Web サービスが含まれます。VDB Editor は VDB の内容やデプロイ可能 (検証) な状態を管理する方法を提供します。

**VDB Editor** には以下のように上部パネルと下部パネルが含まれます。上部パネルには **Models** タブと **Other Files** タブが含まれます。下部パネルには、Data Roles、VDB Description、および Translator Overrides タブが含まれます。



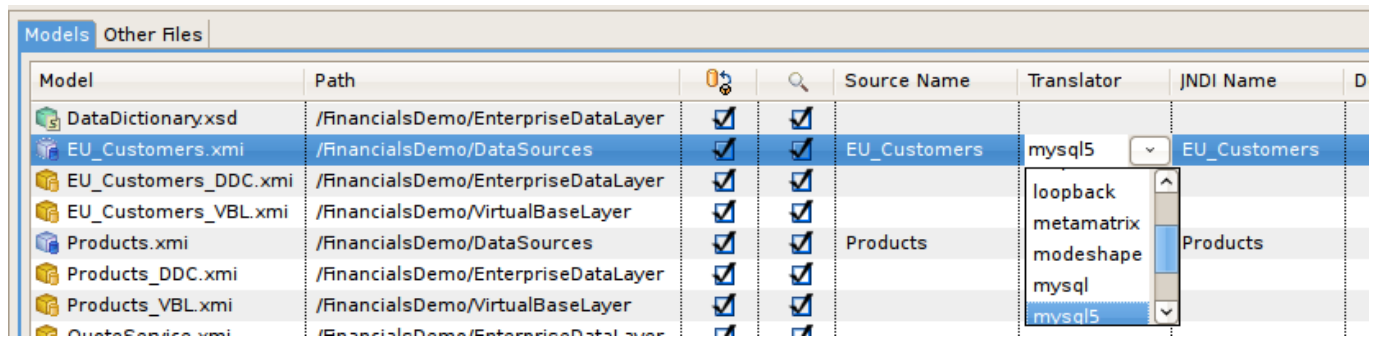
図E.19 VDB Editor

VDB の内容を管理するには、右のボタンでモデルの追加または削除を使用します。

各モデルの Visibility チェックボックスでモデルの表示を設定します。特定モデルおよびそれらのメタデータコンテンツを GUI ツールで公開されたスキーマから削除し、低レベルのデータアクセスセキュリティを提供します。

VDB を完全にクエリー可能にするには、Source Name、Translator、および JNDI Names に有効な値を指定する必要があり、JBoss Data Virtualization サーバーでデプロイされたアーティファクトを表す必要があります。

Deiid Designer ランタイムプラグインがインストールされ、JBoss Data Virtualization サーバーが稼働している場合、VDB エディターでソースモデルを選択し、右クリックで **Change Translator** または **Change JNDI Data Source** を選択すると、サーバーで適用可能なアーティファクトを選択できます。



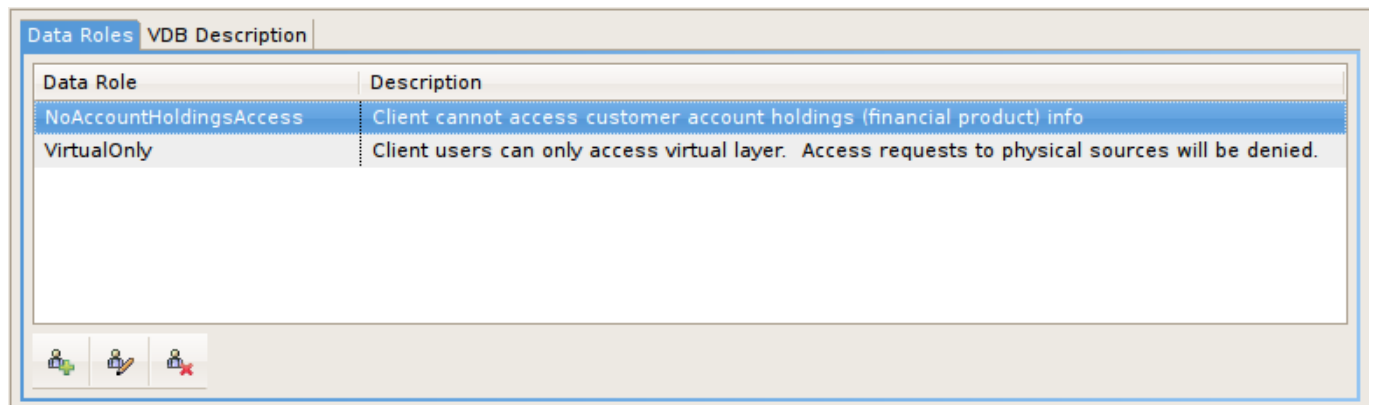
図E.20 トランスレーターまたはデータソースの変更

デフォルトの JBoss Data Virtualization サーバーインスタンスが定義および接続されている場合、トランスレーターおよび JNDI テーブルセルに利用可能なトランスレーターのドロップダウンリストとそのサーバーで利用可能な JNDI 名が含まれます。

## E.8.2. Data Roles の編集

Teiid Designer は VDB 固有のデータロールを作成、編集、および管理する方法を提供します。セキュリティーオプションが有効な状態 (デフォルト) で JBoss Data Virtualization サーバー内にデプロイされると、Teiid JDBC 接続よりこの VDB に対して実行されたクエリーは、VDB のデータロールによって定義されたデータアクセスパーミッションに準拠します。

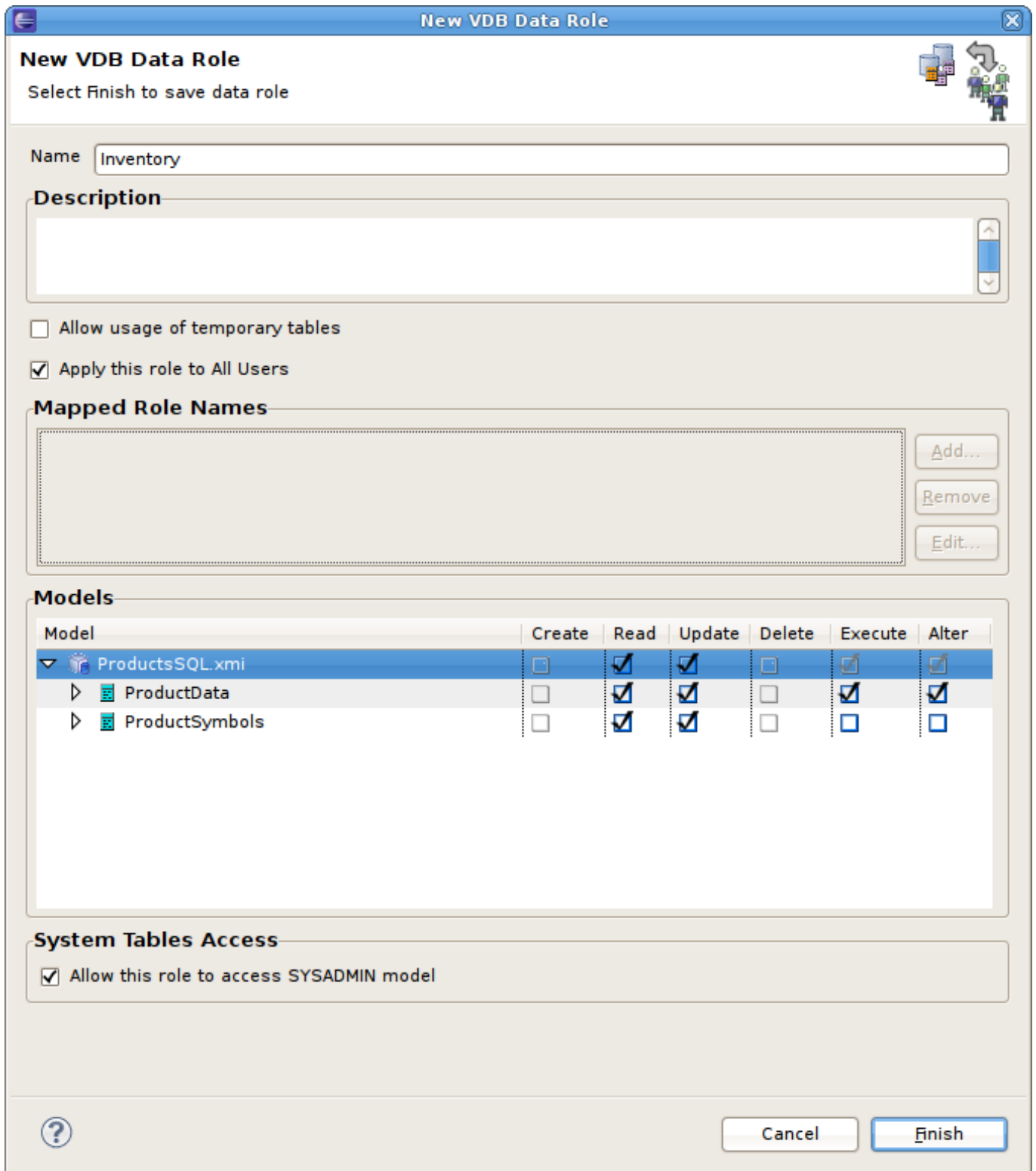
VDB Editor には、**List of current data roles** および **New...**、**Edit...** および **Remove** アクションボタンで構成される **VDB Data Roles** セクションが含まれます。



図E.21 VDB Data Roles パネル

**New...** または **Edit...** をクリックすると **New VDB Data Role** エディターダイアログが表示されます。一意なデータロール名を指定し、説明を任意で追加します。Models セクションのエントリーを選択または選択解除して、各モデル要素の CRUD 値を編集します。



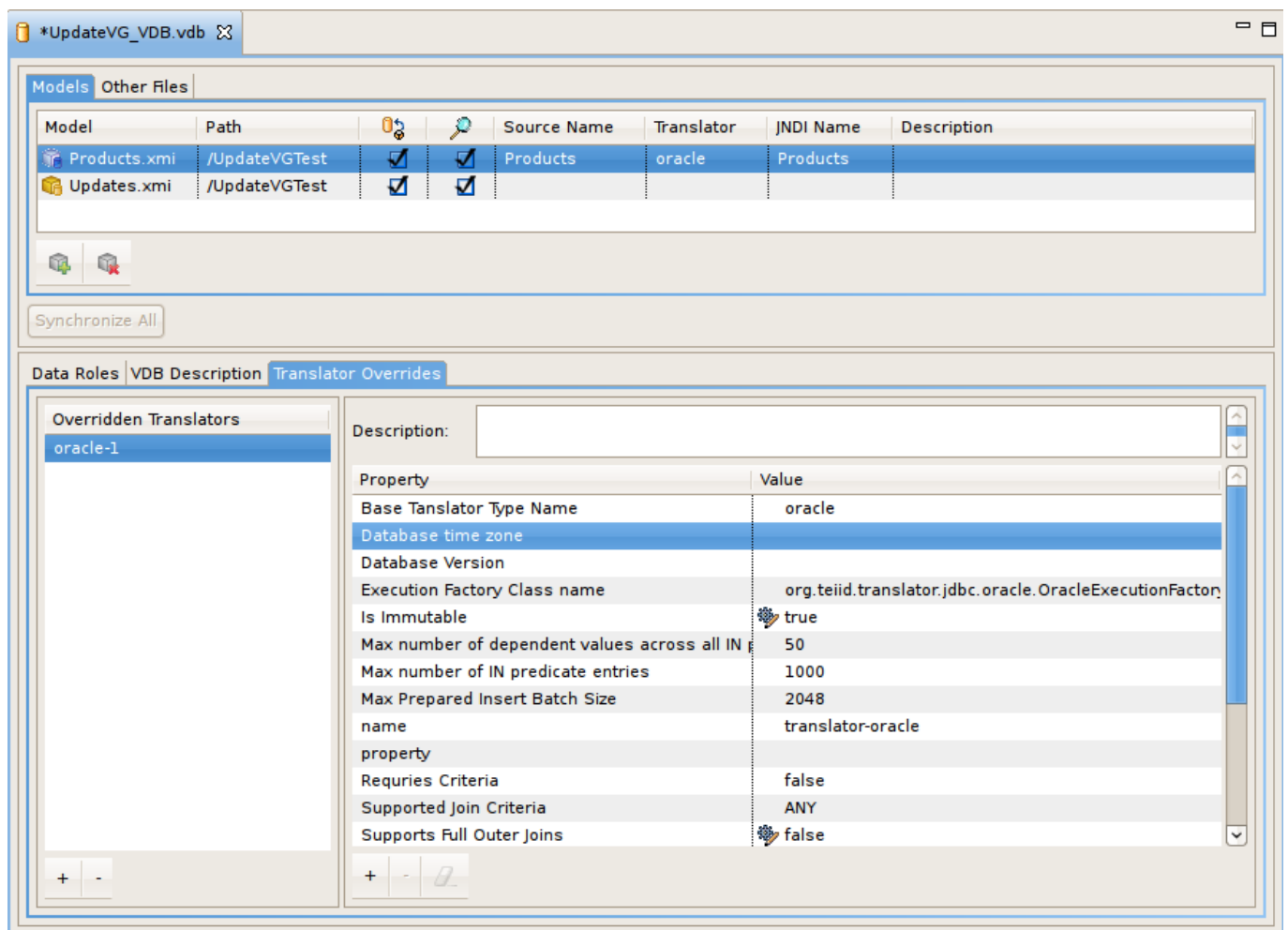


図E.22 VDB Data Roles タブ

### E.8.3. Translator Overrides の編集

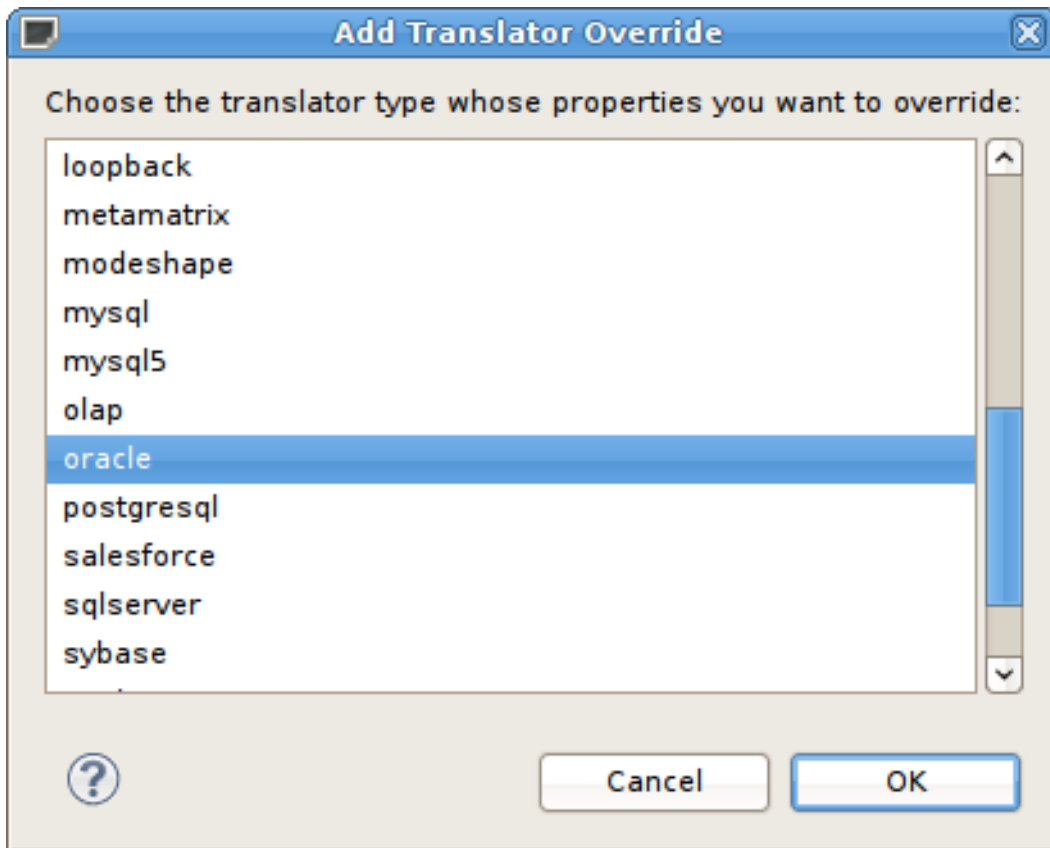
**Teiid Designer** では、**Translator Overrides** タブで VDB に固有のトランスレーターオーバーライドプロパティを作成、編集、および管理できます。トランスレーターオーバーライドは、特定のソースモデルのデータソースを対象とするデフォルトでないプロパティのセットです。そのため、各トランスレーターオーバーライドには oracle、db2、mysql などのターゲットトランスレーター名と、デフォルトでないキーと値のプロパティセットが必要です。

VDB Editor には、左側の **List of current translator overrides**、右側のプロパティエディターパネル、およびパネル下部の **Add (+)** および **Remove (-)** アクションボタンで構成される **Translator Overrides** セクションが含まれます。



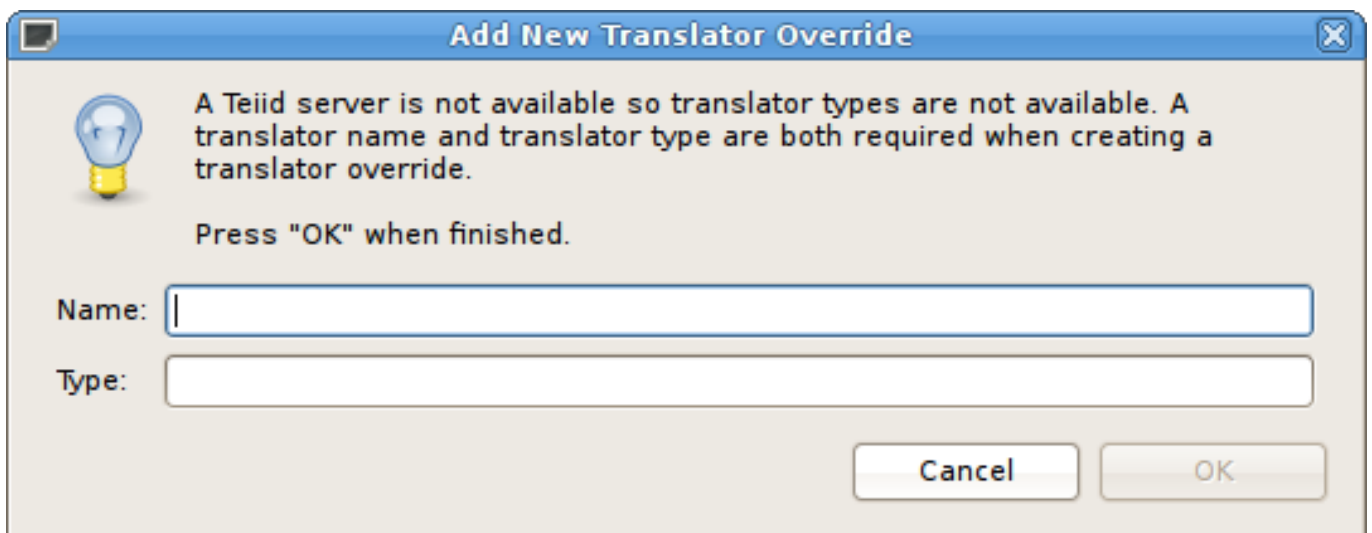
図E.23 VDB Translator Overrides タブ

特定のトランスレータータイプをオーバーライドするには、**(+)** をクリックしてトランスレーターを追加します。デフォルトの JBoss Data Virtualization サーバーインスタンスが接続され、利用可能であると以下の **Add Translator Override** ダイアログが表示されます。既存のトランスレータータイプを選択し、**OK** をクリックします。オーバーライドは VDB 内のソースのみに適用できるため、必ず VDB のソースモデルの 1 つに対応するトランスレータータイプを選択してください。パネル右側のプロパティパネルには、プロパティのデータタイプ (boolean、integer、string など) を基にした各プロパティタイプの編集可能なセルが含まれます。



図E.24 Add Translator Override ダイアログ

デフォルトの Teiid サーバーインスタンスがない場合は、**Add New Translator Override** ダイアログが表示されます。トランスレーターオーバーライドの一意な名前 (例: oracle\_override) と有効なトランスレータータイプ名 (例: oracle) を入力し、**OK** をクリックします。パネルの右側にあるプロパティパネルでは、キーと値の文字列ベースのプロパティセットを追加、編集、および削除します。これらのプロパティを編集するとき、すべての値は型文字列として処理されます。

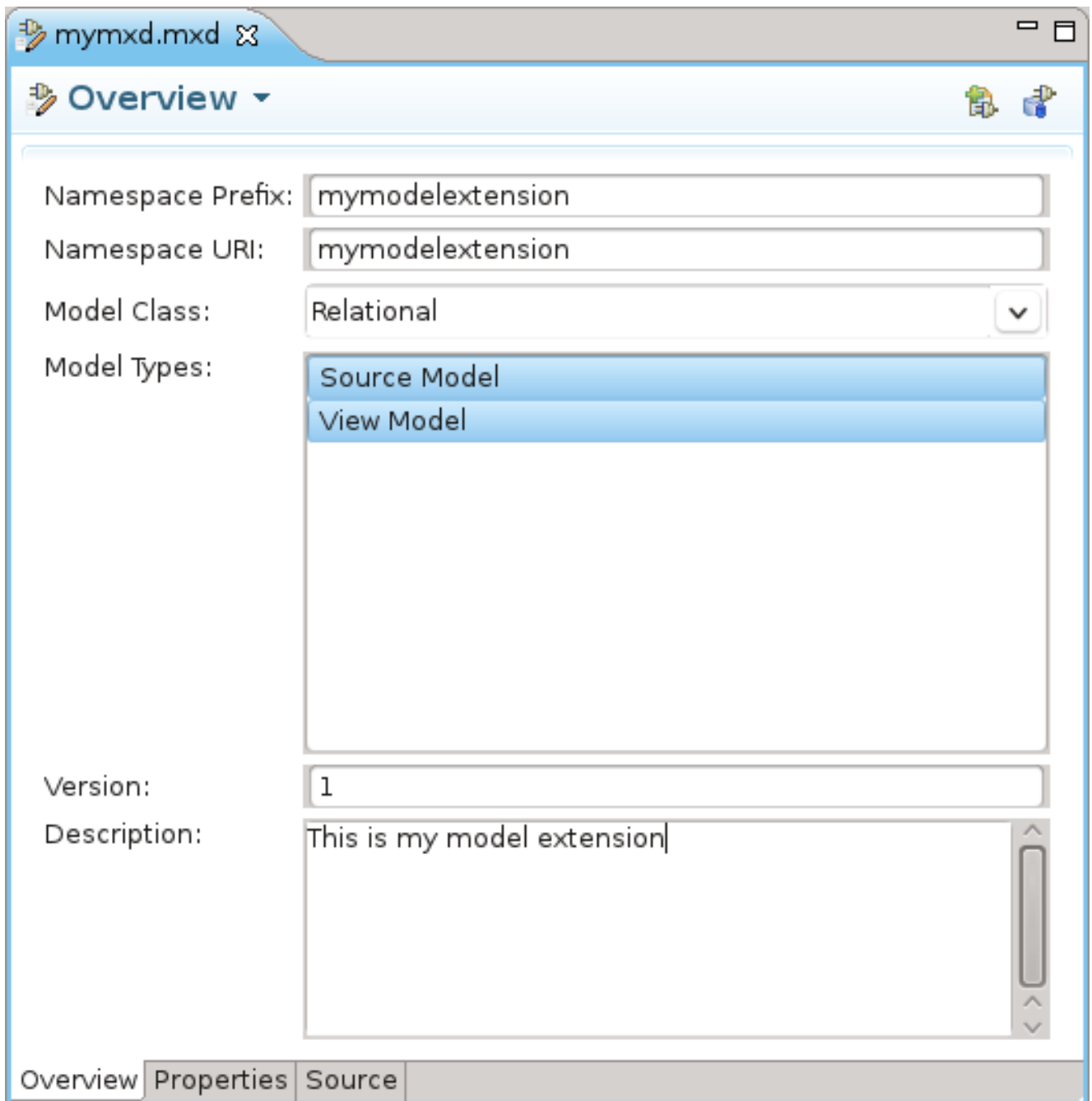


図E.25 Add New Translator Override ダイアログ

## E.9. Model Extension Definition Editor

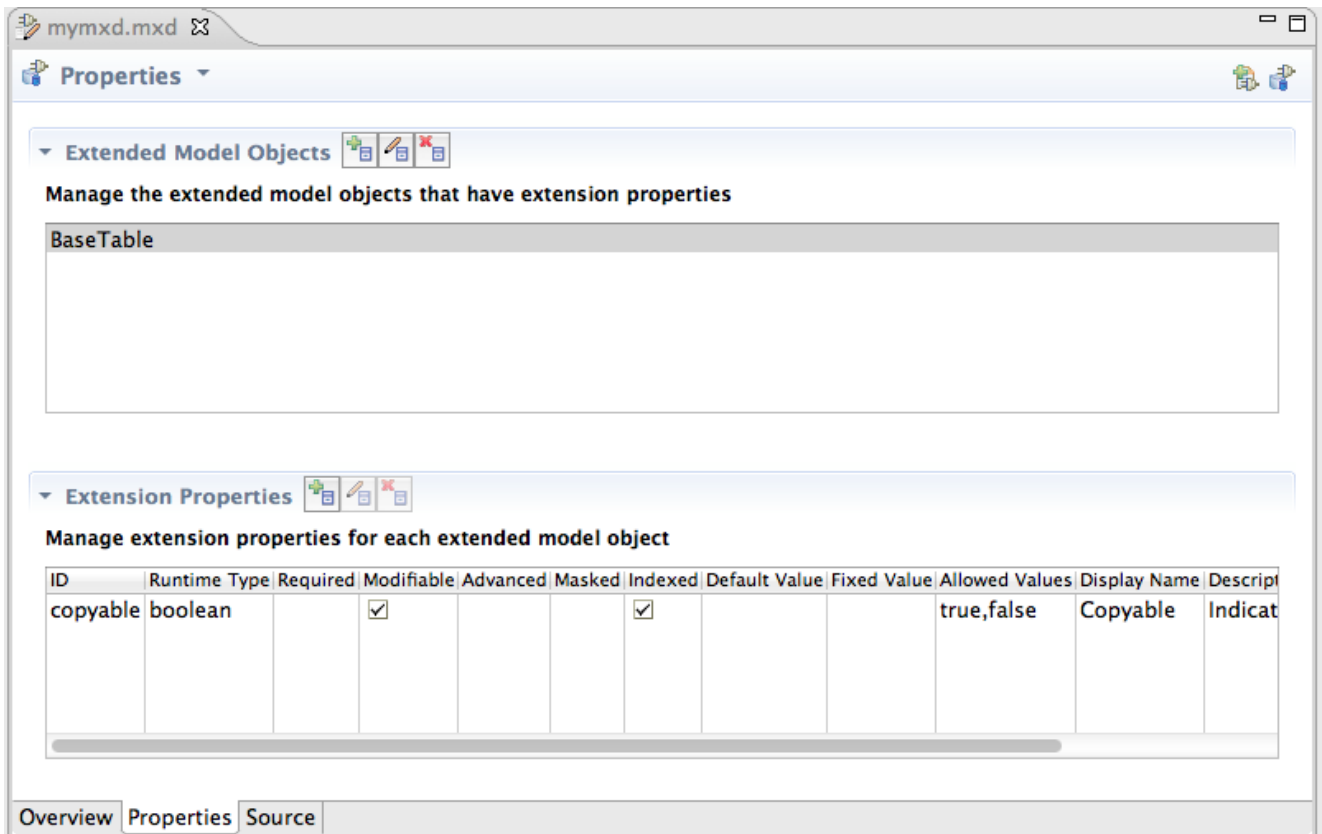
Model Extension Definition (MED) Editor は複数のタブがあるエディターで、ワークスペースのユーザー定義 (\*.mxd ファイル) を作成および編集するために使用されます。MED Editor には、ヘッダーセクションを共有する 3 つのサブエディター (Overview、Properties、および Source) があります。

- ※ Overview サブエディター - このエディターで一般的な MED 情報が管理されます。この情報には、名前空間接頭辞、名前空間 URI、拡張モデルクラス、および説明が含まれます。以下に Overview サブエディターを示します。



図E.26 概要タブ


- ※ Properties サブエディター - このエディターで MED 拡張プロパティが管理されます。各拡張プロパティはモデルオブジェクトタイプに関連付けられる必要があります。Properties サブエディターは 2 つセクションに分割されます (Extended Model Objects および Extension Properties)。以下に Properties サブエディターを示します。



図E.27 Properties タブ

- ※ Source - このタブは MED の詳細を表示する読み取り専用の XML ソースビューアーです。このソースビューアーは編集できません。

GUI コンポーネントのデータに検証エラーが存在する場合、Overview および Properties サブエディターの GUI コンポーネントにエラーアイコンが表示されます。エラーアイコンの上マウスポインターを置くと、ツールヒントとエラーメッセージが表示されます。エラーマークは共通のヘッダーセクションに表示されるエラーメッセージと関連しています。エラーマークの例を以下に示します。

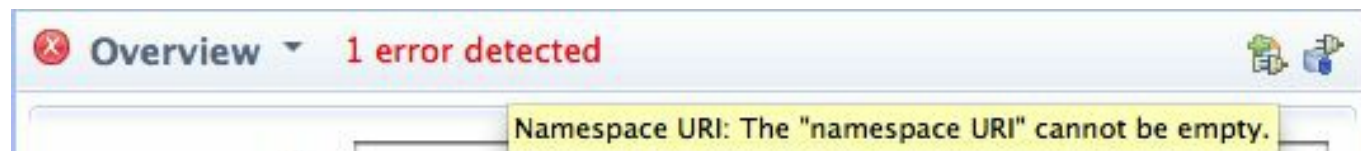
Namespace URI: 

図E.28 エラーのあるテキストフィールド

MED サブエディターはヘッダーセクションを共有します。ヘッダーは以下で構成されます。

- ※ Status Image - 最も重大な検証メッセージ (エラー、警告、または情報) を示すイメージ。検証メッセージがない場合はモデル拡張イメージが表示されます。
- ※ Title - サブエディターのタイトルが表示されます。
- ※ Menu - レジストリーの MED に追加および更新するアクションと、Model Extension Registry ビューを表示するアクションが含まれるドロップダウンメニュー。
- ※ Validation Message - このエリアには OK またはエラー概要メッセージが表示されます。概要メッセージが表示される場合、メッセージのツールヒントはすべてのメッセージを列挙します。
- ※ Toolbar - 同じアクションがドロップダウンメニューとして含まれます。

以下は、エラーメッセージのツールヒントが含まれる共有ヘッダーセクションの例になります。



図E.29 共有ヘッダーの例

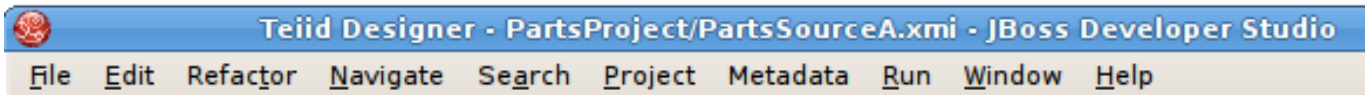
## 付録F Teiid Designer メインメニュー

### F.1. Teiid Designer メインメニュー

Teiid Designer のメインメニューバーには 8 つのカテゴリのアクションがあります。

カテゴリには以下が含まれます。

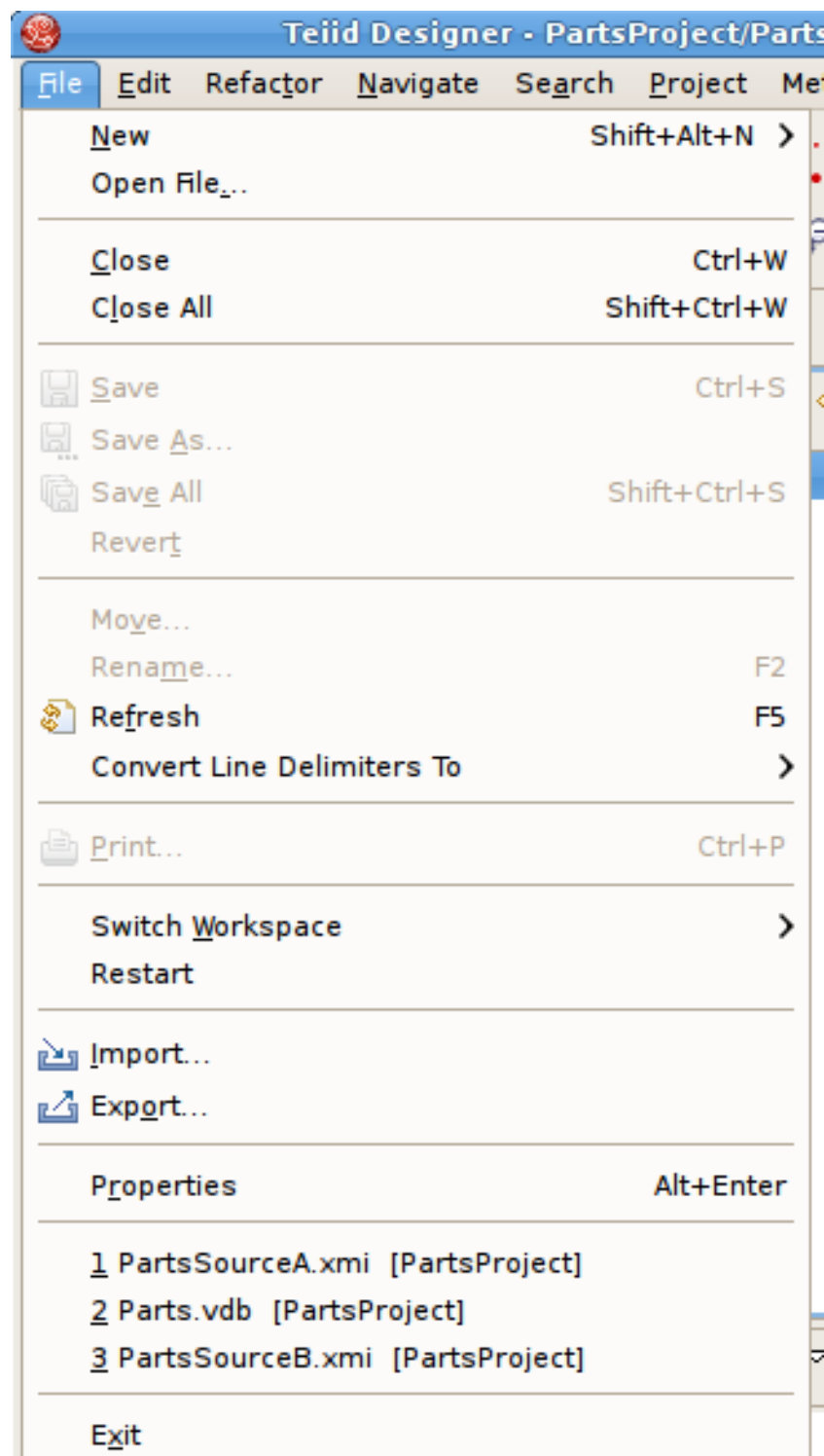
- ※ リソース管理アクション
- ※ undo (元に戻す) と redo (やり直す) が含まれる標準の編集アクション
- ※ リソースアクション (Rename、Move など)
- ※ ワークスペース内でのデータの検索
- ※ モデルレベルアクション
- ※ カスタムメタデータに関するアクション
- ※ パースペクティブの変更、またはパースペクティブへのビューの追加や削除
- ※ 利用できる Teiid Designer ヘルプドキュメント、Teiid Designer SQL Support Guide、および Eclipse の概要へのアクセス



図F.1 アプリケーションメインメニュー

### F.2. File メニュー

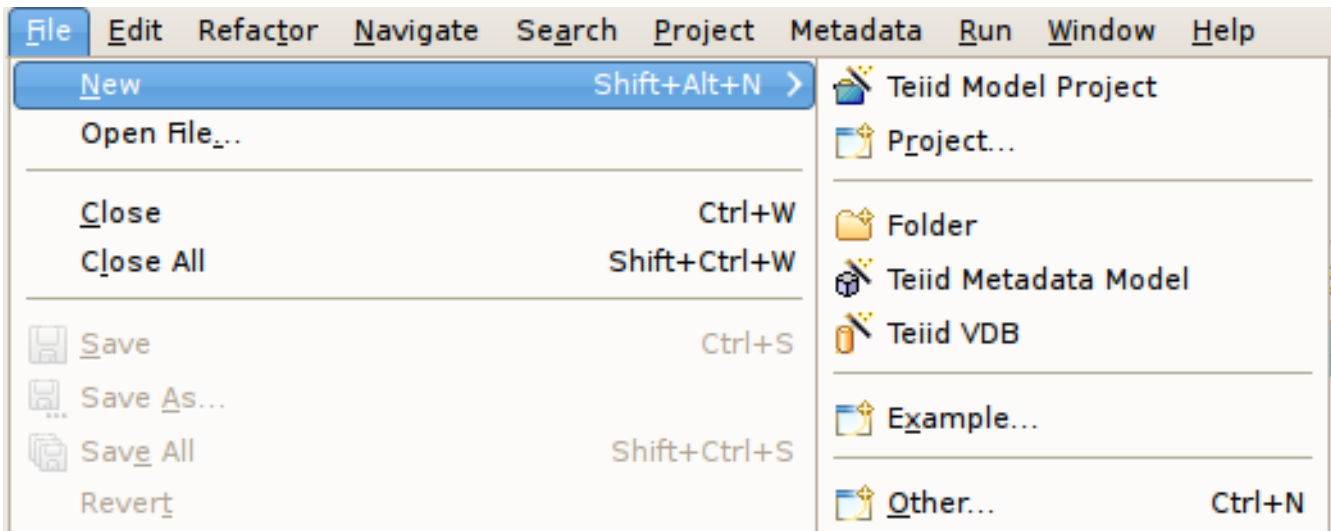
**File** メニューはワークスペースリソースを管理するアクションを提供します。



図F.2 File メニュー

**New** > サブメニューは、さまざまな汎用ワークスペースリソース、Teiid Designer モデル、および VDB を作成するためのアクションを提供します。







図F.3 File メニュー

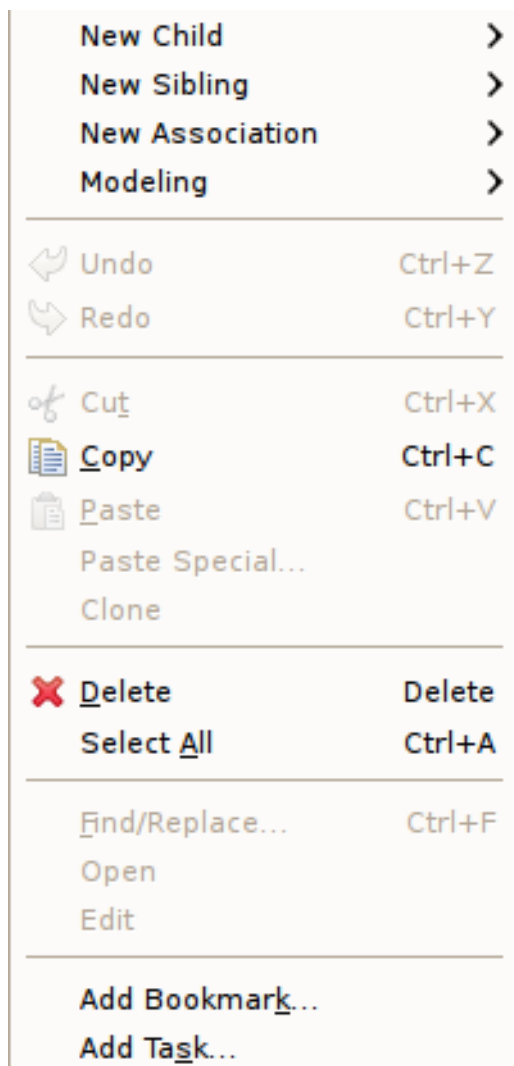
**File** メニューには以下のアクションが含まれます。

- ✧  **New** > Model Project - Create user a new model project.
- ✧  **New** > Folder - Create new folder within an existing project or folder.
- ✧  **New** > Model - Create a new model of a specified model type and class using the New Model Wizards.
- ✧  **New** > Virtual Database Definition - Create a new VDB, or Virtual Database Definition.
- ✧ **Open File** - ファイルを開いて編集することができます (ワークスペースにないファイルを含む)。
- ✧ **Close (Ctrl+W)** - アクティブなエディターを閉じます。ファイルを閉じる前に変更を保存するよう促されます。
- ✧ **Close All (Shift+Ctrl+W)** - 開かれているエディターをすべて閉じます。ファイルを閉じる前に変更を保存するよう促されます。
- ✧  **Save (Ctrl+S)** - Saves the contents of the active editor.
- ✧  **Save As** - Enables you to save the contents of the active editor under another file name or location.
- ✧  **Save All (Shift+Ctrl+S)** - Saves the contents of all open editors.
- ✧ **Move** - Refactor > Move resource ダイアログを起動します。
- ✧ **Rename (F2)** - リソースが選択された場合は Refactor > Rename resource ダイアログが起動されます。その他の場合はインラインの名前変更が実行されます。
- ✧ **Refresh** - ファイルシステムの内容でリソースが更新されます。
- ✧ **Convert Line Delimiters To** - 選択されたファイルの行区切り文字を変更します。変更は即座に反映され、再度区切り文字を変更するまで保持されます。ファイルを保存する必要はありません。
- ✧  **Print (Ctrl+P)** - Prints the contents of the active editor. In the Teiid Designer, this action prints the diagram in the selected editor. Allows control over orientation (portrait or landscape), scaling, margins and page order. User can also specify a subset of the pages to print (i.e., 2 through 8).

- ※ Switch Workspace - 別のワークスペースに切り替えできる Workspace Launcher を開きます。これによりワークベンチが再起動されます。
- ※ Restart - ワークベンチを終了し、再起動します。
- ※  --> Import - Launches the Import Wizard which provides several ways to construct or import models..
- ※  Export - Launches the Export Wizard which provides options for exporting model data.
- ※ Properties (**Alt+Enter**) - 現在選択されているリソースの Properties ダイアログを開きます。これには、ファイルシステムのリソースへのパス、最終変更された日付、および書き込み可能または実行可能な状態が含まれます。
- ※ Most Recent Files List - ワークベンチの最後にアクセスされたファイルのリストが含まれます。ファイル名を選択して File メニューからこれらのファイルを開くことができます。
- ※ Exit - ワークベンチを閉じ、終了します。







### F.3. Edit メニュー

**Edit** メニューは、モデルおよびプロジェクトリソースの内容、構造、およびプロパティを管理するアクションを提供します。以下の図は、メタデータモデルが選択されたときに表示される **Edit** メニューを表しています。



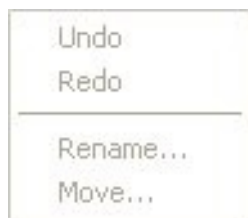
図F.4 Edit メニュー

**Edit** メニューには以下のアクションが含まれます。

- ※ New > Child - このメニューは、選択されたオブジェクト下で作成できるあらゆるタイプの子オブジェクトの作成をサポートするために動的に作成されます。
- ※ New > Sibling - このメニューは、選択されたオブジェクトと同じ親で作成できるあらゆるタイプの兄弟オブジェクトの作成をサポートするために動的に作成されます。
- ※ New > Association - このメニューは、選択されたオブジェクトと作成できるあらゆるタイプの関連 (アソシエーション) の作成をサポートするために動的に作成されます。
- ※ Modeling > - このメニューは動的に作成されます。さまざまなモデリング演算は選択されたモデルオブジェクトタイプを基に表示されます。
- ※  Undo - Reverses the effect of the most recent command.
- ※  Redo - Reapplies the most recently undone command.
- ※  Cut - Deletes the selected object(s) and copies it to the clipboard.
- ※  Copy - Copies the selected object(s) to the clipboard.
- ※  Paste - Pastes the contents of the clipboard to the selected context.
- ※ Paste Special... - 複雑なクリップボードオブジェクトに対して追加の貼り付け機能を提供します。
- ※ Clone - 選択したオブジェクトを同じ場所に同じ名前で作成します。ユーザーはツリーの新しいオブジェクトの名前を変更できます。
- ※  Delete - Deletes the selected object(s).
- ※ Select All - 現在のビューのオブジェクトをすべて選択します。
- ※ Rename - ユーザーがツリーのオブジェクトの名前を変更できるようにします。
- ※ Find/Replace - Transformation Editor など、現在のテキストビューでの検索に使用できるダイアログを起動します。
- ※ Open - 適切なエディターで選択したオブジェクトを開きます。
- ※ Edit - Choice Editor や Recursion Editor など、適切な専門エディターで選択したオブジェクトを開きます。
- ※ Add Bookmark... - このコマンドは、カーソルが現在表示されている行のアクティブなファイルにブックマークを追加します。
- ※ Add Task... - このコマンドは、カーソルが現在表示されている行のアクティブなファイルにタスクを追加します。

## F.4. Refactor メニュー

**Refactor** メニューは、モデルへのファイルレベルの変更に対して Teiid Designer 固有のアクションを提供します。



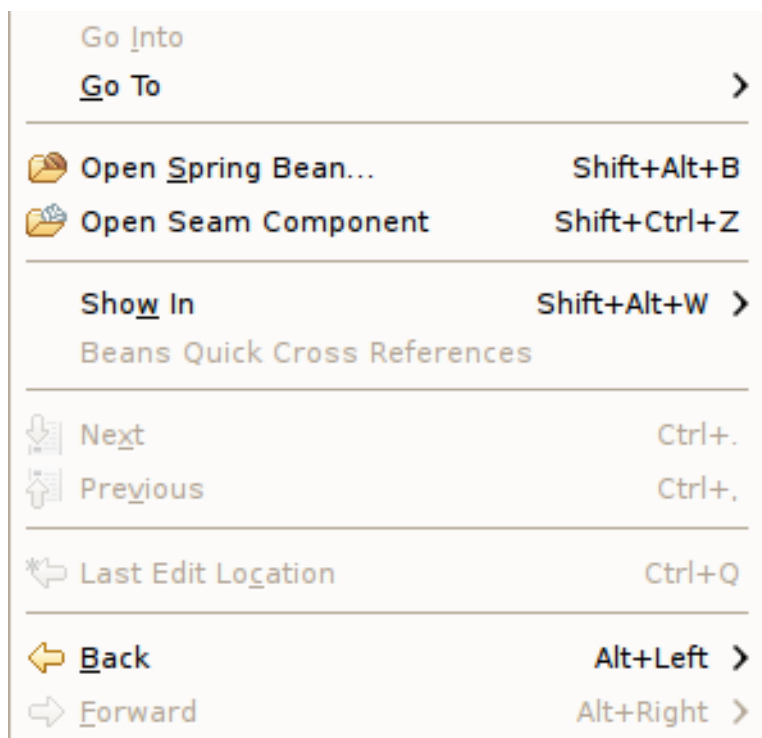
図F.5 Refactor メニュー

The **Refactor** メニューには以下のアクションが含まれます。

- ✦ Undo - 最後に実行した refactor コマンドを取り消します。
- ✦ Redo - 最後に取り消した refactor コマンドを再度実行します。
- ✦ Move - モデルをあるコンテナ (フォルダーまたはプロジェクト) から別のコンテナに移動します。
- ✦ Rename - モデルの名前を変更します。

## F.5. Navigate メニュー

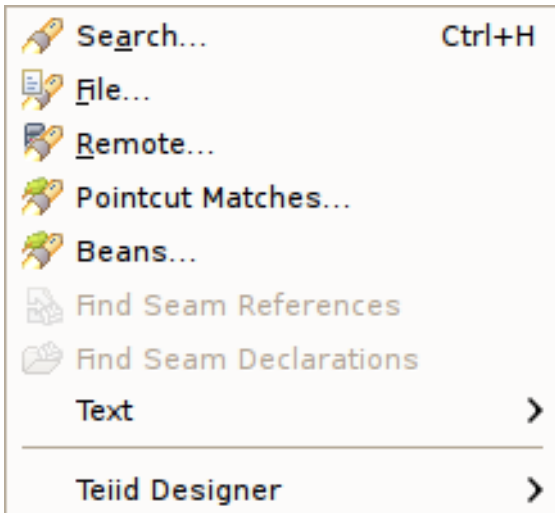
現在、**Teiid Designer** はアクションを **Navigate** メニューに提供しません。詳細は Eclipse のドキュメントを参照してください。



図F.6 Navigate メニュー

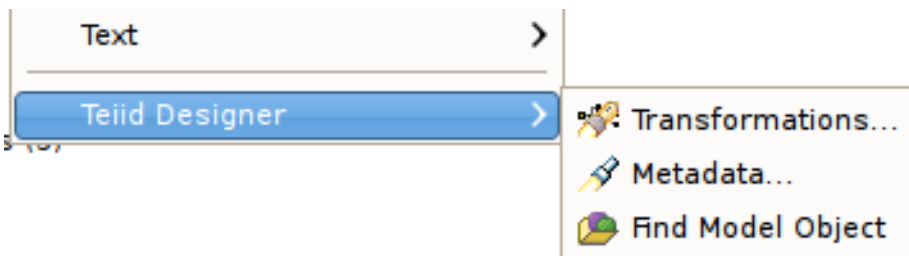
## F.6. Search メニュー

**Search** メニューは複数の検索オプションを提供します。






図F.7 Search メニュー

Teiid Designer は以下のようにサブメニュー (例: Teiid Designer >) をメインの検索メニューに提供します。

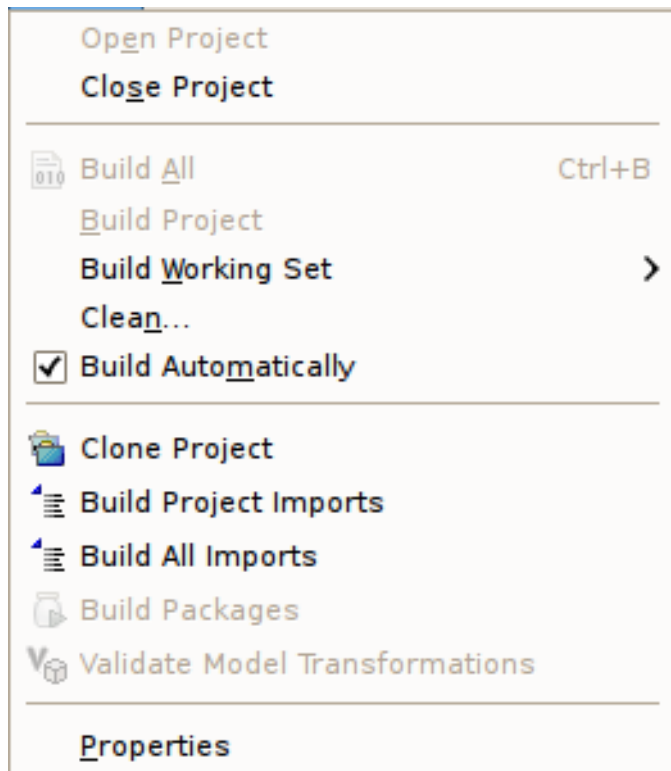


図F.8 Search メニュー

Teiid Designer サブメニューの各アクションの詳細は次のとおりです。


- ※  Transformations... - Launches the Transformation Search dialog. User can search models in the workspace for matching SQL text. Search results appear in the dialog and user can select and view SQL as well as open desired transformations for editing.
- ※  Metadata... - Launches the Search dialog. User can search for models in the workspace by specifying an Object Type, and/or a Data Type, and/or a property value. Search results appear in the Search Results view, and double-clicking a result will open that model in the appropriate editor.
- ※  Find Model Object - Launches the Find Model Object dialog, which can be used to find an object in the workspace by specifying all or part of its name. Selecting the object will open it in the appropriate editor.

## F.7. Project メニュー



図F.9 Project メニュー

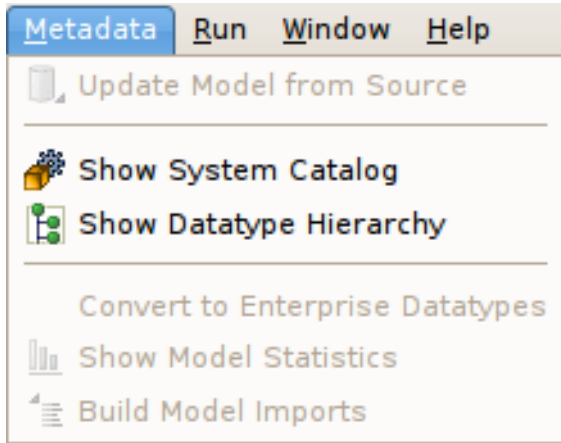
**Project** メニューの各アクションの詳細は次のとおりです。

- ✦ Open Project - Open Project ダイアログを表示します。
- ✦ Close Project - 現在選択されているプロジェクトを閉じます。
- ✦  Build All - Validates the contents of the entire workspace. Any errors or warnings will appear in the Problems View.
- ✦ Build Project - 選択されたプロジェクトの内容を検証します。エラーや警告は Problems ビューに表示されます。
- ✦ Build Working Set - 選択されたワーキングセットの内容を検証します。エラーや警告は Problems ビューに表示されます。
- ✦ Clean.. - Clean ダイアログを表示します。
- ✦ Build Automatically - Build Automatically フラグを on または off に設定します。on の場合、このメニュー項目の左側にチェックマークが表示され、Save が実行されるたびに変更の検証が自動的に行われます。
- ✦ Clone Project - Clone Project ダイアログを表示します。
- ✦ Build Project Imports - 選択されたプロジェクト内に含まれるモデルのモデルインポート依存関係をすべて調整します。
- ✦ Build All Imports - ワークスペース内に含まれるモデルのモデルインポート依存関係をすべて調整します。
- ✦ Build Packages - 現在未定義
- ✦ Validate Model Transformations - 選択されたビューモデルのトランスフォーメーションをすべて再検証します。

- ※ Properties - 選択されたファイルに対するオペレーティングシステムのファイルプロパティダイアログを表示します。

## F.8. Metadata メニュー

**Metadata** メニューは Teiid Designer 固有のアクションを提供します。



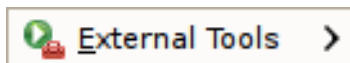
図F.10 Metadata メニュー

**Metadata** メニューには以下のアクションが含まれます。

- ※ Update Model from Source - 選択したモデルが JDBC インポートより作成されたリレーショナルソースモデルである場合、モデルはデータベーススキーマの変更を基に更新されます。
- ※ Show System Catalog - System Catalog ビューを開きます。
- ※ Show Datatype Hierarchy - Datatype Hierarchy ビューを開きます。
- ※ Re-resolve References - 別のモデルコンポーネントへのモデル内の参照を分析します。
- ※ Convert to Enterprise Datatypes - 追加のプロパティを選択されたスキーマモデル内の簡単なデータタイプへ追加し、エンタープライズデータタイプとしてラベル付けします。
- ※ Show Model Statistics - 選択したモデルの Model Statistics ダイアログを開きます。
- ※ Build Model Imports - 選択したモデルのモデルインポート依存関係をすべて調整します。

## F.9. Run メニュー

現在、**Teiid Designer** はアクションを **Run** メニューに提供しません。詳細は Eclipse のドキュメントを参照してください。



図F.11 Window メニュー

## F.10. Window メニュー

以下の **Window** メニューには Teiid Designer 固有のアクションは含まれていません。詳細は Eclipse Workbench ドキュメントを参照してください。



図F.12 Window メニュー

**Preferences...** アクションは、Teiid Designer の機能やデフォルト値を設定できる **Preferences** ダイアログを表示します。

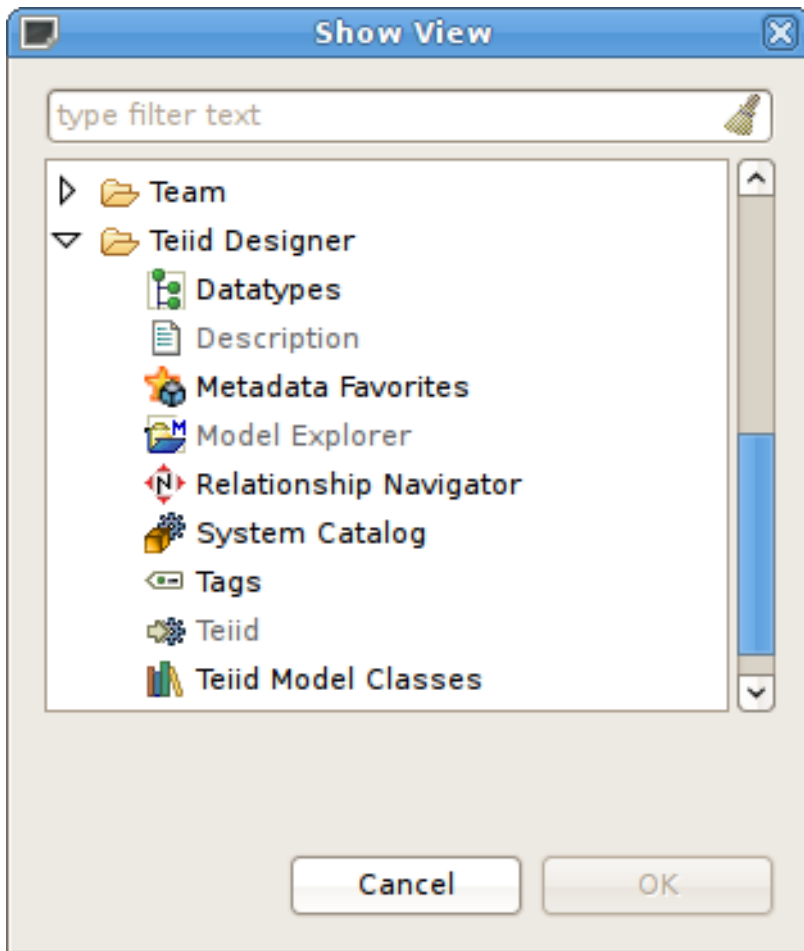


### 注記

これらのメニュー項目は、インストールされた Eclipse 機能およびプラグインのセットによって異なることがあります。

1 つ以上の Teiid Designer ビューが含まれるようにパースペクティブをカスタマイズするには、**Show View > Other...** アクションをクリックし、**Teiid Designer** カテゴリを展開して使用できるビューを表示します。

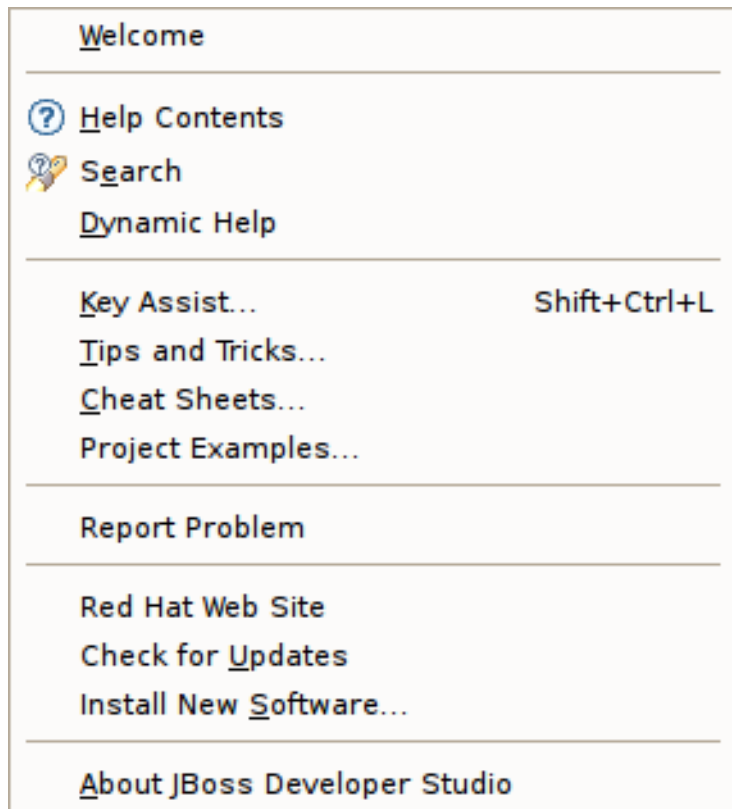




図F.13 Show View ダイアログ

## F.11. Help メニュー

以下の **Help** メニューには Teiid Designer 固有のアクションは含まれていません。詳細は Eclipse Workbench ドキュメントを参照してください。



図F.14 Help メニュー

各アクションの詳細は次のとおりです。

- ※ Welcome - ドキュメント、例、手順の開始点へのリンクが含まれる Welcome パースペクティブへ移動します。
- ※ Help Contents  - Launches the Help Window. All of Designer's online documentation is accessible from there as well.
- ※ Search  - Launches the Help Search view, which can be used to search for phrases in the documentation.
- ※ Dynamic Help - 格納された動的ヘルプビューを開きます。
- ※ Key Assist (**Ctrl-Shift-L**) ... - 既存のキー補助バインディングを説明するダイアログを表示します。
- ※ Tips and Tricks... - Tips and Tricks ヘルプページの 1 つを選択できるダイアログを表示します。
- ※ Cheat Sheets... - Eclipse チートシートの 1 つを選択できるダイアログを表示します。
- ※ Project Examples... - さまざまなプロジェクトサンプルを即座にワークスペースにインポートできる JBoss によるアクション。
- ※ Report Problem - 簡単に問題を報告できる JBoss によるアクション。
- ※ Check for Updates... - インストールされた Eclipse ソフトウェアへの更新を取得できます。
- ※ Install New Software... - 新しいソフトウェアをワークベンチにインストールできます。
- ※ About JBoss Developer Studio - About ダイアログを表示します。

## 付録G 改訂履歴

|   |                        |                        |
|---|------------------------|------------------------|
| <b>改訂 6.3.0-27.4</b><br>翻訳完成                              | <b>Wed Feb 1 2017</b>  | <b>Kenzo Moriguchi</b> |
| <b>改訂 6.3.0-27.3</b><br>翻訳ファイルを XML ソースバージョン 6.3.0-27 と同期 | <b>Wed Feb 1 2017</b>  | <b>Kenzo Moriguchi</b> |
| <b>改訂 6.3.0-27.2</b><br>翻訳ファイルを XML ソースバージョン 6.3.0-27 と同期 | <b>Mon Jan 16 2017</b> | <b>Terry Chuang</b>    |
| <b>改訂 6.3.0-27.1</b><br>翻訳ファイルを XML ソースバージョン 6.3.0-27 と同期 | <b>Mon Jan 16 2017</b> | <b>Terry Chuang</b>    |
| <b>改訂 6.3.0-27</b><br>バージョン 6.3 用に更新。                     | <b>Fri Oct 14 2016</b> | <b>David Le Sage</b>   |
| <b>改訂 6.2.0-03</b><br>6.2 用に更新                            | <b>Fri Nov 20 2015</b> | <b>David Le Sage</b>   |