# **Teiid Designer User Guide**

1

3.2.0.Beta

1. Introduction	1
1.1. What is Teiid Designer?	. 1
1.2. Why Use Teiid Designer?	1
1.3. Metadata Overview	. 2
1.3.1. What is Metadata	. 2
1.3.2. Editing Metadata vs. Editing Data	3
1.3.3. Metadata Models	3
1.3.4. Business and Technical Metadata	. 4
1.3.5. Design-Time and Runtime Metadata	5
1.3.6. Source and View Metadata	. 6
1.4. Models 101	9
1.4.1. What Are Models	9
1.4.2. How is a Model Defined?	11
1.4.3. Model Classes and Types	11
1.4.4. Models and VDBs	12
1.4.5. Models and Connectors	12
1.4.6. Model Validation	12
1.4.7. Testing Your Models	13
2. Teiid Designer Perspectives	14
2.1. Teiid Designer Perspective	16
2.2. SQL Explorer Perspective	17
3. Teiid Designer Main Menu	19
3.1. File Menu	19
3.2. Edit Menu	23
3.3. Refactor Menu	25
3.4. Search Menu	25
3.5. Project Menu	26
3.6. Metadata Menu	28
3.7. Window Menu	29
3.8. Help Menu	29
4. Teiid Designer Views	31
4.1. Model Explorer View	31
4.1.1. Selection-Based Action Menus	33
4.2. Outline View	35
4.2.1. Outline Tree View	35
4.2.2. Outline Thumbnail View	35
4.3. UDFs View	37
4.4. Connectors View	40
4.5. Properties View	42
4.6. Description View	44
4.7. Editors	45
4.8. Problems View	47
4.8.1. Toolbar Items	48
4.8.2. Context Menu	50

	4.9. Message Log View	51
	4.10. Preview Results View	52
	4.11. Search Results View	55
	4.12. Datatype Hierarchy View	56
	4.13. Model Classes View	57
	4.14. System Catalog View	57
	4.15. Database Structure View	58
	4.16. SQL History View	59
	4.17. Connections View	59
	4.18. SQL Editor View	60
	4.19. SQL Results View	60
5. E	ditors	62
	5.1. Model Editor	63
	5.1.1. Diagram Editor	63
	5.1.2. Table Editor	76
	5.1.3. Simple Datatypes Editor	81
	5.1.4. Semantic Editor	82
	5.1.5. Source Editor	82
	5.1.6. Model Object Editors	82
	5.2. VDB Editor	83
	5.2.1. Overview	84
	5.2.2. Problems	85
	5.2.3. Indexes	85
	5.2.4. User Files	86
	5.2.5. Web Services	87
	526 Execute	87
6. In	nporters	89
••••••	6.1 Import Archive File	89
	6.2 Import From File System	91
	6.3 Import From JDBC	92
	6.4. Import Metadata From Text File	98
	6.4.1. Import Relational Tables Text Importer	100
	6.4.2 Import Relational Virtual Tables Text Importer	101
	6.4.3 Import Relationships Text Importer	103
	6.4.4 Import Datatypes Text Importer	105
	6.5 Import WSDL as Relational Source	107
	6.6 Import WSDL Into Web Service	110
	6.6.1 Import WSDI From Workspace Location	111
	6.6.2 Import WSDL From File System Location	118
	6.6.3 Import WSDL From URI	124
	6.7 XML Schema as Relational Source	130
	6.8 XSD Schema File	136
	6.9 Import Rational Rose Model	140
7 N	ew Model Wizards	1 <u>7</u> 0
1.140		- 70

7.1. Creating New Relational Source Model	147
7.1.1. Copy From Existing Model	148
7.1.2. Generate From Existing UML Model	149
7.2. Creating New Relational View Model	154
7.2.1. Copy From Existing Model	155
7.2.2. Transform From Existing Model	156
7.2.3. Generate From Existing UML Model	156
7.2.4. Create From XML Schema	156
7.3. Creating New XML Service Source Model	157
7.3.1. Copy From Existing Model	158
7.4. Creating New XML Service View Model	159
7.4.1. Copy From Existing Model	160
7.5. Creating XML Document View Model	161
7.5.1. Copy From Existing Model	161
7.5.2. Build XML Documents From XML Schema	162
7.6. Creating XML Schema Model	165
7.6.1. Copy From Existing Model	166
7.7. Creating Web Service View Model	166
7.7.1. Copy From Existing Model	167
7.7.2. Build From Existing WSDL File(s) or URL	168
7.7.3. Build From Relational Models	168
7.7.4. Build From XML Document View Models	171
7.8. Creating New Extensions Model	173
7.8.1. Copy From Existing Model	173
7.9. Creating New Relationship Model	174
7.9.1. Copy From Existing Model	175
7.10. Creating New UML 2.0 Model	176
7.10.1. Copy From Existing Model	177
8. Creating and Editing Model Objects	178
8.1. Creating New Model Objects	178
8.1.1. New Child Action	178
8.1.2. New Sibling Action	181
8.1.3. New Association Action	185
8.2. Model Object Editors	189
8.2.1. Transformation Editor	189
8.2.2. Input Set Editor (XML)	201
8.2.3. Choice Editor (XML)	203
8.2.4. Recursion Editor (XML)	206
8.2.5. Operation Editor	209
9. Editing Models and Projects	212
9.1. Rename A Model	212
9.2. Move Model	213
9.3. Save Copy of Model	213
9.4. Clone Project	216

10. Testing Your Models	219
10.1. Manage Connectors	219
10.1.1. Manage Connectors	219
10.1.2. Create New Connector	220
10.1.3. Binding Sources to Connectors	221
10.1.4. Importing Connectors and Types	222
10.1.5. Export Connectors	225
10.2. Previewing Data For a Model	226
10.2.1. Preview Relational Table or View	227
10.2.2. Preview Relational Table With Access Pattern	227
10.2.3. Preview Relational Procedure	228
10.2.4. Preview Web Service Operation	229
10.3. Testing With Your VDB	230
11. Searching	237
11.1. Finding Model Objects	238
11.2. Search Models Via Relationship Properties	239
11.3. Search Transformation SQL	240
11.4. Search Models Via Metadata Properties	242
12. Relationships	244
12.1. Creating Relationships	244
12.2. Creating your own Relationship Type	252
12.3. Relationship Navigator	257
13. User Preferences	263
13.1. Teiid Designer Preferences	263
13.1.1. Diagram Preferences	264
13.1.2. Diagram Printing Preferences	265
13.1.3. Editor Preferences	266
13.1.4. Validation Preferences	269
13.1.5. XML Schema Source Preferences	272
13.2. SQL Explorer Preferences	272
13.2.1. SQL Editor Preferences	273

# Introduction

#### Go To: Table of Contents [index.html]

The Teiid Designer User's Guide provides detailed descriptions of Teiid Designer features and functionality.

# 1.1. What is Teiid Designer?

Teiid Designer is an Eclipse-based graphical modeling tool for modeling, analyzing, integrating and testing multiple data sources to produce Relational, XML and Web Service Views that expose your business data.

Designer - NorthWindProject/Northwind_	<u>\$2K</u>	l.xmi - Eclipse Platform
File Edit Refactor Navigate Search Project Me	tadat	ita Run Window Help
📬 • 🔡 🕼 🗁   😕 😕   🖋   💁 •	1 🔏	<b>? •</b> : <b>↑ •</b> ½ • ½ • ∜• ↔ •
Model Explorer 🛛 📴 Outline 🞼 UDFs 🧮		痛 Northwind_S2K.xmi 🛛 🧻 North
Image: CustomerDemographics		CustomerCus CustomerCus CustomerID : string(5) CustomerTypeID : string F FK_CustomerCustomer F FK_CustomerCustomer P FK_CustomerCustomer P FK_CustomerCustomer P FK_CustomerCustomer
📽 Connectors 🛛 🗖		
🎄 🖄 🖄 👘		
SQL Server JDBC Connector	<ul> <li></li> <li></li> </ul>	Package Diagram     Table Editor
Properties 🛛 📄 Description 🗖		🥵 Message Log 🛛

#### Figure 1.1. Teiid Designer

# 1.2. Why Use Teiid Designer?

Teiid Designer is a visual tool that enables rapid, model-driven definition, integration and testing of data services without programming. With Teiid Designer, not only do you map from data sources to target formats using a visual tool, but you can also:

- · resolve semantic differences
- create virtual data structures at a physical or logical level
- use declarative interfaces to integrate, aggregate, and transform the data on its way from source to a target format which is compatible and optimized for consumption by your applications
- resolve semantic differences

This allows you to abstract the structure of the information you expose to and use in your applications from the underlying physical data structures. With Teiid Designer, data services are defined quickly, the resulting artifacts are easy to maintain and reuse, and all the valuable work and related metadata are saved for later reference.

You can use Teiid Designer to integrate multiple sources, and access them using the common data access standards:

- Web Services / SOAP / XML
- JDBC / SQL
- ODBC / SQ

Teiid Designer is an integral part of the Teiid Designer enterprise-class system for providing data services for service-oriented architectures.

# 1.3. Metadata Overview

#### 1.3.1. What is Metadata

**Metadata** is data about data. A piece of metadata, called a meta object in the Teiid Designer, contains information about a specific information structure, irrespective of whatever individual data fields that may comprise that structure.

Let's use the example of a very basic database, an address book. Within your address book you certainly have a field or column for the ZIP code (or postal code number). Assuming that the address book services addresses within the United States, you can surmise the following about the column or field for the ZIP code:

- Named ZIPCode
- Numeric
- A string
- Nine characters long
- Located in the StreetAddress table
- Comprised of two parts: The first five digits represent the five ZIP code numbers, the final four represent the ZIP Plus Four digits if available, or 0000 if not

• Formatted only in integer numeric characters. Errors will result if formatted as 631410.00 or 6314q0000

This definition represents metadata about the ZIP code data in the address book database. It abstracts information from the database itself and becomes useful to describe the content of your enterprise information systems and to determine how a column in one enterprise information source relates to another, and how those two columns could be used together for a new purpose

You can think of this metadata in several contexts:

- What information does the metadata contain? (see **Business and Technical Metadata**)
- What data does the metadata represent? (see Source and View Metadata)
- How will my organization use and manage this metadata? (see <u>Design-Time and Runtime</u> <u>Metadata</u>)

# 1.3.2. Editing Metadata vs. Editing Data

The Teiid Designer helps you to create and describe an abstract graphic representation of your data structure of your data in the original data sources. It also describes whether those data sources are composed of Relational databases, text files, data streams, legacy database systems, or some other information type.

The Teiid Designer allows you to create, edit, and link these graphically-represented meta objects that are really a description of your data, and not the data itself.

So when this documentation describes the process of creating, deleting, or editing these meta objects, **remember** that you are not, in fact, modifying the underlying data.

## 1.3.3. Metadata Models

A **metadata model** represents a collection of metadata information that describes a complete structure of data.

In a previous example we described the field ZIPCode as a **metadata object** in an address book database.. This **meta object** represents a single distinct bit of metadata information. We alluded to its parent table, StreetAddress. These **meta objects**, and others that would describe the other tables and columns within the database, would all combine to form a **Source Metadata** model for whichever enterprise information system hosts all the objects.

You can have **Source Models** within your collection of **metadata models** These model physical data storage locations. You can also have **View Models**, which model the business view of the data. Each contains one type of metadata or another. For more information about difference between Source and View metadata, see "Source and View Metadata."

NOTE: For detailed information about creating models from your metadata, see Models 101

# **1.3.4. Business and Technical Metadata**

Metadata can include different types of information about a piece of data.

- **Technical metadata** describes the information required to access the data, such as where the data resides or the structure of the data in its native environment.
- Business metadata details other information about the data, such as keywords related to the meta object or notes about the meta object.

Note that the terms **technical and business metadata**, refer to the content of the metadata, namely what type of information is contained in the metadata. Don't confuse these with the terms "physical" and "view" metadata that indicate what the metadata represents. For more information, see Source and View Metadata.

## 1.3.4.1. Technical Metadata

Technical metadata represents information that describes how to access the data in its original native data storage. Technical metadata includes things such as datatype, the name of the data in the enterprise information system, and other information that describes the way the native enterprise information system identifies the meta object

Using our example of an address book database, the following represent the technical metadata we know about the ZIP code column:

- Named ZIPCode
- Nine characters long
- A string
- Located in the StreetAddress table
- Uses SQL Query Language

These bits of information describe the data and information required to access and process the data in the enterprise information system.

#### 1.3.4.2. Business Metadata

Business metadata represents additional information about a piece of data, not necessarily related to its physical storage in the enterprise information system or data access requirements. It can also represent descriptions, business rules, and other additional information about a piece of data.

Continuing with our example of the ZIP Code column in the address book database, the following represents business metadata we may know about the ZIP code:

• The first five characters represent the five ZIP code numbers, the final four represent the ZIP Plus Four digits if available, or 0000 if not

• The application used to populate this field in the database strictly enforces the integrity of the data format

Although the first might seem technical, it does not directly relate to the physical storage of the data. It represents a business rule applied to the contents of the column, not the contents themselves.

The second, of course, represents some business information about the way the column was populated. This information, although useful to associate with our definition of the column, does not reflect the physical storage of the data.

## 1.3.5. Design-Time and Runtime Metadata

Teiid Designer software distinguishes between design-time metadata and run-time metadata. This distinction becomes important if you use the Teiid Designer Server. Design-time data is laden with details and representations that help the user understand and efficiently organize metadata. Much of that detail is unnecessary to the underlying system that runs the Virtual Database that you will create. Any information that is not absolutely necessary to running the Virtual Database is stripped out of the run-time metadata to ensure maximum system performance.

## 1.3.5.1. Design-Time Metadata

Design-time metadata refers to data within your local directory that you have created or have imported. You can model this metadata in the Teiid Designer, adding **Source** and **View** metadata.

## 1.3.5.2. Runtime Metadata

Once you have adequately modeled your enterprise information systems, including the necessary technical metadata that describes the physical structure of your sources, you can use the metadata for data access.

To prepare the metadata for use in the Teiid Designer Server, you take a snapshot of a metadata model for the Teiid Designer Server to use when resolving queries from your client applications. This run-time metadata represents a static version of design-time metadata you created or imported. This snapshot is in the form of a **Virtual Database** definition, or **VDB**.

As you create this runtime metadata, the Teiid Designer:

- · derives the runtime metadata from a consistent set of metadata models.
- creates a subset of design-time metadata, focusing on the technical metadata that describes the access to underlying enterprise information systems.
- optimizes runtime metadata for data access performance.

You can continue to work with the design-time metadata, but once you have created a runtime metadata model, it remains static.

# 1.3.6. Source and View Metadata

In addition to the distinction between business and technical metadata, you should know the difference between **Source Metadata and View Metadata**.

#### Source and View metadata refer to what the metadata represents, not its content.

**Source Metadata** directly represents metadata for an enterprise information system and captures exactly where and how the data is maintained. Source Metadata sounds similar to technical metadata, but Source Metadata can contain both technical and business metadata. When you model Source Metadata, you are modeling the data that your enterprise information systems contain. For more information, see "Modeling Your Teild Designer Information Systems."

View Metadata, on the other hand, represent tailored views that **transform** the **Source Metadata** into the terminology and domain of different applications. **View Metadata**, too, can contain both technical and business metadata. When you model **View Metadata**, you're modeling the data as your applications (and your enterprise) ultimately use it. For more information, see "**Modeling Your Teiid Designer Data Needs**."

### 1.3.6.1. Modeling Your Source Metadata

When you model the **Source Metadata** within your enterprise information systems, you capture some detailed information, including:

- Identification of datatype
- Storage formats
- Constraints
- Source-specific locations and names

The **Source Metadata** captures this detailed technical metadata to provide a map of the data, the location of the data, and how you access it.

This collection of **Source Metadata** comprises a direct mapping of the information sources within your enterprise. If you use the Teiid Designer Server for information integration, this technical metadata plays an integral part in query resolution.

For example, our ZIPCode column and its parent table StreetAddress map directly to fields within our hypothetical address book database.

To extend our example, we might have a second source of information, a comma-separated text file provided by a marketing research vendor. This text file can supply additional demographic information based upon address or ZIP code. This text file would represent another EIS, and the meta objects in its Source Model would describe each comma-separated value.

## 1.3.6.2. Modeling Your View Metadata

When you create **View Metadata**, you are not describing the nature of your physical data storage. Instead, you describe the way your enterprise uses the information in its day-to-day operations. **View Metadata** derives its classes and attributes from other metadata. You can derive **View Metadata** from **Source Metadata** that describes the ultimate sources for the metadata or even from other View Metadata. However, when you model **View Metadata**, you create special "views" on your existing enterprise information systems that you can tailor to your business use or application expectations. This **View Metadata** offers many benefits:

- You can expose only the information relevant to an application. The application uses this **View Metadata** to resolve its queries to the ultimate physical data storage.
- You can add content to existing applications that require different views of the data by adding the **View Metadata** to the existing **View Metadata** that application uses. You save time and effort since you do not have to create new models nor modify your existing applications.
- Your applications do not need to refer to specific physical enterprise information systems, offering flexibility and interchangeability. As you change sources for information, you do not have to change your end applications.
- The **View Metadata** models document the various ways your enterprise uses the information and the different terminology that refers to that information. They do so in a central location.

Our example enterprise information sources, the address book database, and the vendor-supplied comma-delimited text file, reside in two different native storage formats and therefore have two Source Metadata models. However, they can represent one business need: a pool of addresses for a mass mailing.

By creating a **View Metadata** model, we could accurately show that this single View Table, the AddressPool, contains information from the two enterprise information systems. The **View Metadata** model not only shows from where it gets the information, but also the SQL operations it performs to select its information from its source models.

This **View Metadata** can not only reflect and describe how your organization uses that information, but, if your enterprise uses the Teiid Designer Server, your applications can use the **View Metadata** to resolve queries.

To create this **View Metadata**, you create a view and define a **transformation** for that view, a special query that enables you to select information from the source (or even other view) metadata models. For more information, see "*Transformation Editor*."

#### 1.3.6.3. Modeling Metadata Transformations

Section on Modeling transformations.

#### 1.3.6.3.1. Metadata Transformations

By modeling View Metadata, you can illustrate the business view of your enterprise information sources. View Metadata models not only describe that business view, but also illustrate how the meta objects within the View Metadata models derive their information from other metadata models.

Let's return to the example of our address book database and the vendor's comma-separated list. We want to generate the View Metadata model, Address Pool, from these enterprise information systems.



#### Figure 1.2. Data Flow for View Transformations

The transformation that joins these metadata models to create the virtual Address Pool metadata model contains a SQL query, called a union, that determines what information to draw from the source metadata and what to do with it.

The resulting Address Pool contains not only the address information from our Address Book database, but also that from our vendor-supplied text file.

#### 1.3.6.3.2. SQL in Transformations

Transformations contain SQL queries that SELECT the appropriate attributes from the information sources.

For example, from the sources the transformation could select relevant address columns, including first name, last name, street address, city, state, and ZIP code. Although the metadata models could contain other columns and tables, such as phone number, fax number, e-mail address, and Web URL, the transformation acts as a filter and populates the Address Pool metadata model with only the data essential to building our Address Pool.

You can add other SQL logic to the transformation query to transform the data information. For example, the address book database uses a nine-character string that represents the ZIP Plus

Four. The transformation could perform any SQL-supported logic upon the ZIPCode column to substring this information into the format we want for the Address Pool View metadata model.

#### 1.3.6.3.3. Mapping XML Transformations

When you model View Metadata, you can also create a View XML Document model. This View Document lets you select information from within your other data sources, just like a regular View Metadata model, but you can also map the results to tags within an XML document.



#### Figure 1.3. Data Flow for XML Transformations

In this example, the Address Pool View Metadata model still selects its information from the Address Book Database and the Vendor Text File, but it also maps the resulting columns into tags in the Address XML document.

# 1.4. Models 101

## 1.4.1. What Are Models

A model is a representation of a set of information constructs. A familiar model is the relational model, which defines tables composed of columns and containing records of data. Another familiar model is the XML model, which defines hierarchical data sets.

In Teiid Designer, models are used to define the entities, and relationships between those entities, required to fully define the integration of information sets so that they may be accessed in a

uniform manner, using a single API and access protocol. The file extension used for these models is '.xmi' (Example: NorthwindOracle.xmi) which adheres to the XMI syntax defined by the OMG.

Below is an example of the partial contents of a model file.

<pre><?xml version="1.0" encoding="ASCII"?> <xml <="" <mmcore:modelannotation="" description="Northwind_Oracle was cre (modelImports xmi:uuid=" mmuuid:2d915780-4140-1079-9d18-8acf4a7127d5"="" mmuuid:2e663940-4140-1079-9d18-8acf4a7127d5"="" modellocation="" name="SimpleDatatypes-instance" pre="" version="2.0" xm="" xmi:uuid="mmuuid:b0355f00-413b-1079-9d18-8acf4a7127d5" xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></xml></pre>
<pre>//www.core.indefinition/ //websitional-BaceTable wminuid-bone64c0_413b_1079_9d18_8acf4a7127d5" name="CATECOPIES" nameTaSources"CATECOP</pre>
<pre>(columns xm::uuid="mmuuid:bb5ac3c0-413b-1079-9418-8acf4a7127d5" name="CATEGORVID" nameInSource="CATEGORVID" native</pre>
<pre><columns na<="" name="CATEGORYNAME" nameinsource="CATEGORYNAME" td="" xm1:uuld="mmuuld:bo4ee/cU-413b-1079-9d18-8act4a/127d5"></columns></pre>
<pre><columns name="DESCRIPTION" nameinsource="DESCRIPTION" nat;<="" td="" xml:uuld="mmuuld:bo4ee/cl=413b=1079-9d18=8act4a/127d5"></columns></pre>
<pre></pre> <pre>&lt;</pre>
<pre><type href="http://www.metamatrix.com/metamodels/SimpleDatatypes-instance#blob"></type> </pre>
<pre></pre> <pre>(rolation)</pre> <pre>// contains/ </pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>

#### Figure 1.4. Sample Model File

Model files should never be modified "by hand". While it is possible to do so, there is the possibility that you may corrupt the file such that it cannot be used within the Teiid Designer system.

The fundamental models in Teiid Designer define the structural and data characteristics of the information contained in data sources. These are referred to as source models (represented by

#### T,

). Teiid Designer uses the information in source models to federate the information in multiple sources, so that from a user's viewpoint these all appear to be in a single source.



#### Figure 1.5. Model Internals

In addition to source models, Teiid Designer provides the ability to define а variety of view models(represented by



). These can be used to define a layer of abstraction above the physical (or source) layer, so that information can be presented to end users and consuming applications in business terms rather than as it is physically stored. Views are mapped to sources using transformations between models. These business views can be in a variety of forms:

- Relational Tables and Views
- XML
- XML Service
- Web services
- Relationships
- Functions
- UML 2.0

A third model type, logical, provides the ability to define models from a logical or structural perspective.

# 1.4.2. How is a Model Defined?

Models are defined using Teiid Designer in various ways:

- Created via importing source data characteristics. (see Import Wizard)
- Manual creation via New Model Wizard
- Transforming or copying from one model into another (see <u>New Model Wizard</u> options)
- · Various custom actions

## 1.4.3. Model Classes and Types

Teiid Designer can be used to model a variety of classes of models. Each of these represent a conceptually different classification of models.

- **Relational** Model data that can be represented in table columns and records form. Relational models can represent structures found in relational databases, spreadsheets, text files, or simple Web services.
- XML Model that represents the basic structures of XML documents. These can be "backed" by XML Schemas. XML models represent nested structures, including recursive hierarchies.
- XML Schema W3C standard for formally defining the structure and constraints of XML documents, as well as the datatypes defining permissible values in XML documents.

- XML Service TBD.
- Web Services which define Web service interfaces, operations, and operation input and output parameters (in the form of XML Schemas).
- Model Extensions for defining property name/value extensions to other model classes.
- Function The Function metamodel supports the capability to provide user-defined functions, including binary source jars, to use in custom transformation SQL statements. Currently only a single UDF model is supported by Teiid Designer. See <u>UDFs View</u> for details.
- Relationship TBD.
- UML 2.0 TBD.

VDBs contain two primary varieties of model types - source and view. Source models represent the structure and characteristics of physical data sources, whereas view models represent the structure and characteristics of abstract structures you want to expose to your applications.

### 1.4.4. Models and VDBs

Models used for data integration are packaged into a virtual database (VDB). The models must be in a complete and consistent state when used for data integration. That is, the VDB must contain all the models and all resources they depend upon. Models contained within a VDB can be imported into the Teiid Designer. In this way, VDBs can be used as a way to exchange a set of related models. (See description in <u>VDB Editor</u> section)

## 1.4.5. Models and Connectors

A connector provides the connectivity to the source for the query engine when it is executing queries to that source.

All Source models must have connectors associated with them in a VDB in order to be tested in Teiid Designer or deployed for data access.

It is possible that multiple models may use the same connector, but each model must have a connector.

In Teiid Designer, connector are automatically created "under the hood" when you import from a specific supported data source. You can also create and maintain your own custom connectors. (see <u>Manage Connectors</u>)

## 1.4.6. Model Validation

Models must be in a valid state in order to be used for data access. Validation of a single model means that it must be in a self-consistent and complete state, meaning that there are no "missing pieces" and no references to non-existent entities. Validation of multiple models checks that all inter-model dependencies are present and resolvable.

Models must always be validated when they are deployed in a VDB for data access purposes.

Enterprise Teiid Designer will automatically validate your models whenever the user Saves (Note: the "Models > Validate Automatically" option must be checked). When editing models, the editor tabs will display a "\*" to indicate that the model has unsaved changes.

# **1.4.7. Testing Your Models**

Designing and working with data is often much easier when you can see the information you're working with. The Teiid Designer's **Preview Data** feature makes this possible and allows you to instantly preview the information described by any object, whether it's a physical table or a virtual view. In other words, you can test the views with actual data by simply selecting the table, view, procedure or XML document. The preview functionality leverages an embedded version of the Teiid Designer Server, so you can be sure that the behavior in the Teiid Designer will reliably match when the VDB is deployed to the Server. Previewing information is a fast and easy way to sample the data. Of course, to run more complicated queries like what your application likely uses, simply execute the VDB in the Teiid Designer and type in any query or SQL statement.

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 <u>*Preview*</u> results view. This action is accessible throughout the Teiid Designer in various view toolbars and context menus.

There are two requirements for previewing your data: the selected object must be one several previewable model object types and all source models within the model dependency tree must be bound to connectors.

Previewable objects include:

- Relational table or view, including tables involving access patterns.
- Relational procedure.
- Web Service operation.
- XML Document staging table.

Note that any virtual table, view or procedure is previewable as long as all "physical" source models are bound via connectors. (See <u>Connectors</u> view and <u>Manage Connectors</u>)



#### Note

If attempting to preview a relational access patter, a web service operation or a relational procedure with input parameters, a dialog will be presented to request values for required parameters.

# **Teiid Designer Perspectives**

#### Go To: Table of Contents [index.html]

Teiid Designer utilizes <u>*Eclipse's*</u> [http://www.eclipse.org] Workbench environment which controls visual layout via perspectives. A perspective defines the initial set and layout of views and editors. Within the application window, each perspective shares the same set of editors. Each perspective provides a set of functionality aimed at accomplishing a specific set of tasks.

Perspectives also control what appears in certain menus and toolbars. They define visible action sets, which you can change to customize a perspective. You can save a perspective that you build in this manner, making your own custom perspective that you can open again later.

Teiid Designer perspectives include:

- Teiid Designer Default perspective providing fundamental model editing and management capability.
- **SQL Explorer** Provides views and actions to perform tests on your VDB by connecting to your data sources and executing your test queries.

For more details on perspectives, views and other Eclipse workbench details, see formal <u>Eclipse</u> <u>Documentation</u> [http://help.eclipse.org/ganymede/index.jsp].

# **Opening a Perspective**

There are two ways to open a perspective:

Using the Open Perspective button

on the shortcut bar.

- Choosing a perspective from the Window > Open Perspective menu.
- To search for string values in your transformations SQL:
  - Step 1 Click on the Open Perspective button
  - Step 2 A menu appears showing the same choices as shown on the Window > Open Perspective menu. Choose Other from the menu.



#### Figure 2.1. Perspectives Menu

• Step 3 - In the Select Perspective dialog choose Teiid Designer and click OK.

🙅 Select Perspective		×
Designer (default) MR MetaMatrix Repository Resource SQLExplorer		_
	OK Cancel	

#### Figure 2.2. Select Perspective Dialog

The Teiid Designer perspective is now displayed.

There are few additional features of perspectives to take note of.

• The title of the window will indicate which perspective is in use.



#### Figure 2.3. Workbench Window Title Bar

- The shortcut bar may contain multiple perspectives. The perspective button which is pressed in, indicates that it is the current perspective.
- To display the full name of the perspectives, right click the perspective bar and select **Show Text** and conversely select **Hide Text** to only show icons.
- To quickly switch between open perspectives, select the desired perspective button. Notice that the set of views is different for each of the perspectives.



#### Figure 2.4. Workbench Window Title Bar

# **2.1. Teiid Designer Perspective**

The Teiid Designer perspective includes several main UI components or groups of components as shown below. They include:

- <u>Model Explorer</u>
- Outline
- <u>UDFs</u>
- <u>Connectors View</u>
- Properties View
- **Description**
- <u>Model Editors</u>

- Designer - Books SourceA.xmi - MetaMatrix Ente	rprise Designer _ 🗆 🛪
<u>File E</u> dit Refac <u>t</u> or <u>N</u> avigate Search <u>P</u> roject <u>M</u> et	adata <u>W</u> indow <u>H</u> elp
[ <sup>1</sup> • ]] [ <sup>1</sup> ] [ <sup>1</sup>	
Model Explorer & Outline UDFs -	📬 Books_SourceA.xmi 🕱 🔭 🗖 🗖
🛒 🎝 🍪 🗘 🖓 🖾	(c)
▼ 🎦 Books Project	
👂 🗁 data	
BookDatatypes.xsd	
▼ 📬 Books_SourceA.xmi	
♦ <sup>4</sup> import declarations (1)	HORS ALASTNAME : string(255)
뭡 Package Diagram	MIDDLEINIT : string(255)
▶ 🖬 AUTHORS	P PK_AUTHORS
	!
🐗 Connectors 🕱 🛛 🖓 🛅 🖻 🗖	
Apache Derby Embedded Connector	▼
Apache Derby Network Connector	
⊕ DB2 JDBC Connector     ▼	Package Diagram Table Editor
Properties 🛛 Description 🔚 🌞 🗔 🏹 🗖 🗖	🖹 Problems 🕱 Message Log 🗖 🗖
Property Value	0 errors, 12 warnings, 0 infos 🛛 💥 🕂
▶ Info	Description Resource
▼ Misc	🚯 The VDB model 'BooksXML.xmi' is BooksWe
Cardinality 🖽 59	▲ The VirtualDatabase 'BooksWebSe BooksWe
Logical Relatic	▲ The VDR model 'RooksYMI vmi' is RooksYN
Materialized TE false	
BaseTable: Books_SourceA/AUTHORS	

# Figure 2.5. Teiid Designer Perspective Layout

# **2.2. SQL Explorer Perspective**

The SQL Explorer perspective includes 4 main UI components or groups of components as shown below. They include:

- Database Structure View
- <u>SQL History View</u>
- Section 4.17, "Connections View"
- SQL Results View

💀 SQLExplorer - PartsSupplier_VDB.vdb - MetaMatrix	k Enterprise De	signer							
File Edit Navigate Search Project Window Help									
🖆 • 🔛 🕼 🏝   👹   🕭   🖋   🏠 🏷 🗘 • न	> -						🗳 🙀		
🗄 Database Structure View 🛛 🖓 🗖	👫 PartsSupplier	_VDB ×	PartsSupplier_	Sour 🧃	PartSupplier_Sour	<b>()</b> F	artsSupplier_	VDB	
PartsSupplier_VDB.vdb-13:34:49 Aug 03	* 🗁 🖉								
	belect SHI "PartsSup	PPER_ID, SHI plier_VDB"."	PPER_NAME PartSuppli	from er_Source)	B.SHIP_VIA"				<
PartSupplier_SourceB.PARTS  PartSupplier_SourceB.SHIP_VIA		M			. (	-		-	
PartSupplier_Sources.SUPPLIER PartSupplier_SourceB.SUPPLIER PartSupplier_SourceB.SUPPLIER_PARTS PartsVirtual.SupplierInfo	1 2 <u>SHIPPER_ID</u> 30 20 10	SHIPPER_NAME DHL Federal Express UPS							
Columns Indexes Primary Key Foreign Key Previs () Colu Data Size Deci Defa Acce SHIPP short 2 0 VES SHIPP string 30 0 NO Column String 30 0 NO									
E SQL History 🕅									
ID         Query           1         select PART_ID, PART_NAME, PART_COLOR, PART_WEIGHT           2         select SHIPPER_ID, SHIPPER_NAME from "PartsSupplier_VDE									
<									
La Connections 🕱 Message Log									
<ul> <li>I Active Session</li> <li>PartsSupplier_VDB.vdb-13:34:49 Aug 03</li> </ul>									
		1	Writable	Insert	1:1				

Figure 2.6. SQL Explorer Perspective Layout

# **Teiid Designer Main Menu**

#### Go To: Table of Contents [index.html]

There are 8 categories of actions on Teiid Designer's main menu bar.

- These categories include:
  - File Resource management actions.
  - *Edit* Standard edit actions including undo/redo.
  - **<u>Refactor</u>** Resource actions (i.e. Rename, Move, etc...).
  - <u>Search</u> Find data within your workspace.
  - **Project** Model level actions.
  - Metadata Custom metadata-related actions.
  - <u>*Window*</u> Change perspectives or add/remove views to your perspective.
  - <u>Help</u> Access available Teiid Designer help documents, Teiid Designer SQL Support Guide and Eclipse Overview information.



#### Figure 3.1. Application Main Menu

# 3.1. File Menu

The File menu provides actions to manage your workspace resources.

New	Alt+Shift+N
🕋 Add Existing Project	
🕋 Add Existing Model Project Se	t
Open File	
Close	Ctrl+W
Close All	Ctrl+Shift+W
Save	Ctrl+S
📓 Save As	
🕞 Save All	Ctrl+Shift+S
Revert	
Move	
Rename	F2
Refresh	F5
Convert Line Delimiters To	
🖻 Print	Ctrl+P
Switch Workspace	
🔄 Import	
🛃 Export	
Properties	Alt+Enter
1 PartsSupplier_SourceA.xmi	[Parts Pr]
2 BooksDoc.xmi [Test_A]	
3 BooksXQuery.xmi [Test_A]	
4 BooksVDB.vdb [Test_A]	
Exit	

### Figure 3.2. File Menu

The **New >** sub-menu provides specific actions to create various generic workspace resources as well as Teiid Designer models and VDBs.



#### Figure 3.3. File Menu

- The File menu contains the following actions:
  - 💣

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 <u>New Model</u> <u>Wizard</u>.

## Ň

**New > Virtual Database Definition** - Create a new VDB, or Virtual Database Definition.

#### • 📬

Add Existing Project... - Add an existing Teiid Designer project into your workspace.

• 📬

Add Existing Model Project Set... - Import an existing Model Project Set from your file system. A Model Project Setan be exported via the Export action.

- Open File Enables you to open a file for editing including files that do not reside in the Workspace.
- Close (Ctrl+W) Closes the active editor. You are prompted to save changes before the file closes.
- Close All (Shift+Ctrl+W) Closes all open editors. You are prompted to save changes before the files close.

## • 📳

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.

- **Revert** Replaces the contents of the active editor with the previously saved contents. Currently not implemented for Teiid Designer models or VDBs.
- Move... Launches a Refactor > Move resource dialog...
- **Rename... (F2)** Launches a Refactor > Rename resource dialog if resource selected, else in-line rename is preformed.
- Refresh Refreshes the resource with the contents in the file system.
- **Convert Line Delimiters To** Alters the line delimiters for the selected files. Changes are immediate and persist until you change the delimiter again you do not need to save the file.
- 🖻

**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 Opens the Workspace Launcher, from which you can switch to a different workspace. This restarts the Workbench.
- Restart Exits and restarts the Workbench.

**Import** - Launches the **Import Wizard** which provides several ways to construct or import models.

## ് പ്

Export - Launches the Export Wizard which provides options for exporting model data.

- **Properties (Alt+Enter)** Opens the **Properties** dialog for the currently selected resource. These will include path to the resource on the file system, date of last modification and its writable or executable state.
- Most Recent Files List Contains a list of the most recently accessed files in the Workbench. You can open any of these files from the File menu by simply selecting the file name.

<sup>•</sup> ès

• Exit - Closes and exits the Workbench.

# 3.2. Edit Menu

The **Edit** menu provides actions to manage the content, structure and properties of your model and project resources.

New Child	
New Sibling	
New Association	
Modeling	
💛 Undo Rename SUPPLIER_PARTS	Ctrl+Z
💫 Redo	Ctrl+Y
f Cut	Ctrl+X
📄 Сору	Ctrl+C
🔁 Paste	Ctrl+V
Paste Special	
Clone	
💢 Delete	Delete
Rename	
Select All	Ctrl+A
Find/Replace	Ctrl+F
Open	Ctrl+C
Edit	
Add Bookmark	
Add Task	

#### Figure 3.4. Edit Menu

- The Edit menu contains the following actions:
  - New > Child This menu is created dynamically to support the creation of whatever types of child objects can be created under the selected object.
  - New > Sibling This menu is created dynamically to support the creation of whatever types of sibling objects can be created under the same parent as the selected object

- New > Association This menu is created dynamically to support the creation of whatever types of associations can be created with the selected object.
- Modeling > This menu is created dynamically. Various modeling operations are presented based on selected model object type.
- 🤣

Undo - Reverses the effect of the most recent command.

• 🍤

Redo - Reapplies the most recently undone command.

· of

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... Provides additional paste capabilities for complex clipboard objects.
- **Clone** Duplicates the selected object in the same location with the same name. User is able to rename the new object right in the tree.
- 🗙

**Delete** - Deletes the selected object(s).

- Rename Allows a user to rename an object in the tree.
- Find/Replace Launches dialog that can be used to search in the current text view, such as a Transformation Editor.
- Open Opens the selected object in the appropriate editor.
- Edit Opens the selected object in the appropriate specialized editor, such as the Choice Editor or Recursion Editor..
- Add Bookmark... This command adds a bookmark in the active file on the line where the cursor is currently displayed.
- Add Task... This command adds a task in the active file on the line where the cursor is currently displayed.

# 3.3. Refactor Menu

The **Refactor** menu provides Teiid Designer specific actions for file-level changes to the models.



### Figure 3.5. Refactor Menu

- The Refactor menu contains the following actions:
  - Undo Undo the last refactor command.
  - Redo Redo the last undone refactor command.
  - Move Move a model from one container (folder or project) to another.
  - Rename Rename a model.

# 3.4. Search Menu

The Search menu presents several specific search options.



#### Figure 3.6. Search Menu

• The **Search** menu contains the following actions:

## • 🔗

Search... - Launches the standard Search dialog, which can be used to find resource

containing specific text or expressions and filter based on file name patterns. For standard Eclipse installations, the default search tile will be File Search.

- 13

**File...** - Launches the standard Search dialog, which can be used to find resource containing specific text or expressions and filter based on file name patterns. The File Search tab will be selected in the dialog.

່ 🔎

**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.

## • 🛃

**Relationship...** - Launches the Search dialog and auto-selects the Relationships tab. User can search for models in the workspace by specifying a relationship type, participant locations, and/or names containing sprcified text. Search results appear in the <u>Search</u> <u>Results</u> view, and double-clicking a result will open that model in the appropriate editor.

• 🚀

**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 <u>Search Results</u> view, and double-clicking a result will open that model in the appropriate editor.

# 3.5. Project Menu

The Project Menu menu ...



Figure 3.7. Project Menu

- The Project menu contains the following actions:
  - Open Project Launches the Open Project dialog.
  - Close Project Closes the currently selected project(s).

# • 📄

**Build All** - Validates the contents of the entire workspace. Any errors or warnings will appear in the **Problems View**.

- **Build Project** Validates the contents of the selected project(s). Any errors or warnings will appear in the **Problems View**.
- Build Working Set Validates the contents of the selected working set. Any errors or warnings will appear in the Problems View.
- Clean.. Launches the Clean dialog.
- **Build Automatically** Sets the **Build Automatically** flag on or off. When on, a check-mark appears to the left of this menu item. When this is turned on, validation of changes is done automatically each time a **Save** is done.
- Validate Model Transformations Revalidates all transformations for the selected view model.
- Clone Project Launches the Clone Project dialog.
- **Build Project Imports** Reconciles all model import dependencies for models contained within the selected project.

- **Build All Imports** Reconciles all model import dependencies for models contained within the workspace.
- Properties Displays the operating system's file properties dialog for the selected file.

# 3.6. Metadata Menu

The Metadata menu provides Teiid Designer-specific actions.



#### Figure 3.8. Metadata Menu

- The Metadata menu contains the following actions:
  - Rebuild VDB Performs a complete re-validation of the contents of the selected VDB.
  - Import... Launches Import Wizard.
  - Update Model from Source If the selected model is a relational source model that was originally created via JDBC Import, then the model will be updated based on changes in the database schema.
  - Export... Launches Export Wizard.
  - Open in Relationship Navigator Opens the Relationship Navigator view and displays the direct relationships to the selected object.
  - Show System Catalog Opens the <u>System Catalog View</u>.

- Show Datatype Hierarchy Opens the Datatype Hierarchy View.
- Re-resolve References Analyzes references within models to other model components.
- **Convert to Enterprise Datatypes** Adds an additional property to simple datatypes within your selected schema model to label them as enterprise datatypes.
- Show Model Statistics Opens the Model Statistics dialog for the selected model.
- Build Model Imports Reconciles all model import dependencies for the selected model.

# 3.7. Window Menu

The **Window** menu shown below contains no Teiid Designer-specific actions. See Eclipse Workbench documentation for details.

New Window	
New Editor	
Open Perspective	•
Show View	ŀ
Customize Perspective.	
Save Perspective As	
Reset Perspective	
Close Perspective	
Close All Perspectives	
Navigation	
Preferences	

#### Figure 3.9. Window Menu

# 3.8. Help Menu

The **Help Menu** menu menu shown below contains no Teiid Designer-specific actions. See Eclipse Workbench documentation for details.

🦚 Welcome	
(?) Help Contents	
💖 Search	
Dynamic Help	
Key Assist	Ctrl+Shift+L
Tips and Tricks	
Software Updates	•
Build Diagnostics Report	
About MetaMatrix Enterprise [	Designer

Figure 3.10. Help Menu

# **Teiid Designer Views**

#### Go To: Table of Contents [index.html]

Teiid Designer provides various views to display and allow editing of models and model content. This section describes each view in detail.

# 4.1. Model Explorer View

Teiid Designer allows you manage multiple projects containing multiple models and any corresponding or dependent resources. The **Model Explorer** provides a simple file-structured view of these resources.

The Model Explorer (shown below) is comprised of a toolbar and a tree view.



Figure 4.1. Model Explorer View

The toolbar consists of nine common actions:
# • 록

Preview Data - Executes a simple preview query (SELECT \* FROM ).

### • ta

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**

The additional actions are shown in the following figure:



Figure 4.2. Additional Actions

If **Show Model Imports** is checked, the imports will be displayed directly under a model resource as shown below.



### Figure 4.3. Show Model Imports Action

### 4.1.1. Selection-Based Action Menus

Selecting specific objects in the **Model Explorer** provides a context from which the Teiid Designer presents a customized menu of available actions.

Selecting a **view model**, for instance, results in a number of high-level options to manage edit model content, perform various operations and provides quick access to other important actions available in Teiid Designer. These may include specialized actions based on model type.



Figure 4.4. Sample Context Menu

# 4.2. Outline View

The Outline View is a utility view which provides both at tree view dedicated to a specific model (open in an editor) and a scaled thumbnail diagram representative of the diagram open in the corresponding Diagram Editor.

You can show the Outline View by clicking on its tab. If there is no open editors, the view indicates that "An Outline is not available." If a Model Editor is open, then the root of the displayed tree will be the model for the editor that is currently in focus in Teild Designer (tab on top).

### 4.2.1. Outline Tree View

This tree view provides the same basic editing and navigation behavior as the Model Explorer. One additional capability is the drag and drop feature which provides re-ordering and re-parenting of objects in a model.



### Figure 4.5. Outline View

### 4.2.2. Outline Thumbnail View

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

# ť,

at the top of the view. The diagram overview displays in the Outline View.

Model Explorer 🔛 Outline 🗙	
🗗 🛱	ж 🤣
	Control of the second sec
A sum	
Numerous Numerous	
Num         Num           1         Num           2         Num           2         Num           2         Num           3         Num	
Annual content of the second sec	

### Figure 4.6. Outline View

The view contains a thumbnail of your entire diagram. The shaded portion represents the portion visible in the Diagram Editor view.

To move to a specific portion of your diagram, click the shaded area and drag to the position you want displayed in the Diagram Editor view.



# 4.3. UDFs View

**User Defined Functions** extend the basic set of functions you can use within data access applications that use Teiid Designer or even within your transformation queries that you use to model View Metadata. Your organization needs to develop Java logic to perform these functions and manage these functions through the Teiid Designer Console. For more information about this complete process, see Creating Custom User-Defined Functions.

Before you can model the user-defined functions, you must have exported the **User Defined Functions** metadata model from Teiid Designer using the Teiid Designer Console. You should have placed this file in your local directory so that you can open it.

The **User Defined Functions** model uses a special metamodel, the Function metamodel, which is limited to constructs for this specific purpose. Therefore you cannot create a new model using this metamodel, nor should you use this model in a project or as a source for View Metadata models.

Users can edit their **User Defined Functions** model from within the UDFs view shown below. The basic functionality of the **UDFs** view is identical to the **Model Explorer** view but focused the single FunctionsDefinitions model.



### Figure 4.7. UDFs View

The view's toolbar actions include:

### |a<sub>z</sub>

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.

### • 🔏

Import UDF Extension Jar -Import your custom UDF extension jar files.

### · 🔂

Add Connector Jar to UDF - Add connector jar to UDF model. This allows using dual use extension jars.

### • 🖻

Collapse All - Collapses the tree view.

### · 🔁

Link with Editor - When object is selected in an open editor, this option auto-selects and reveals object in Model Explorer.

You can delete your UDF jars by right-click select Delete Jars action.



### Figure 4.8. Delete UDF Jars

Both UDF jars and connector extension jars are utilized by Teiid Designer during VDB execution and data Preview. This provides the ability to utilize a common extension jar to deliver both user defined function code and custom connector code. To facilitate this scenerio, the UDFs view contains an Add Connector Jar to UDF action.

To add an existing connector jar to your UDF:

Step 1: Select the Add Connector Jar to UDF toolbar button

or select the UDF jars folder or jar and right-click select the same action.

Step 2 : In the selection dialog below, select one or more connector jars and choose OK to finish.

🙅 Add Connector Jars To UDF	
Select one or more connector jars Select OK to finish	
<ul> <li>axis.jar</li> <li>commons-codec-1.2.jar</li> <li>commons-httpclient-3.0.1.jar</li> <li>commons-logging.jar</li> <li>ConnectorCommons.jar</li> <li>derbyclient.jar</li> <li>jaxen-1.1.jar</li> <li>jdom.jar</li> <li>jms-1.1.jar</li> <li>junk_1.jar</li> <li>postgresql-jdbc.jar</li> <li>sampleconn.jar</li> <li>uddi4j.jar</li> <li>xmlconn.jar</li> <li>XMLConnectorFramework.jar</li> </ul>	
ок	Cancel

### Figure 4.9. Add Connector Jars Dialog

# **4.4. Connectors View**

The **Connectors View** provides a means to display and manage (add or remove) connectors and source bindings in Teild Designer. For more information on Connectors, see <u>Manage</u> <u>Connectors</u>.

To show the **Connectors View** click **Views > Show View > Other...** to display the **Show View** dialog. Choose **Teiid Designer > Connectors** view and hit **OK**.



#### Figure 4.10. Connectors View

The view's toolbar actions include:

### • ऄ

Create New Connector - Launches New Connector wizard.

### • 🖄

Import Connectors - Imports connectors from a \*.cdk file.

### • 🖄

**Export Connectors** - Exports selected connectors to \*.cdk file. This action provides a means to transfer your source connection information between Teiid Designer workspaces.

### • 4

Show/Hide Connector Types - Shows or hides connector types in tree view.

#### • 🖪

Collapse All - Collapses the tree view.

You can also access these actions via the right-click menu as shown below. Note that the **Clone** connector action creates a new connector whose property values are set with those of the originally selected connector.

		à
E 🎲 Books O9i Co	🐝 New	
Books_C Books SQL C Books_C	Edit Clone	
Books SQL C	Delete	
PartsO.> PartsSup PartSup	遒 Import Connectors (*.cdk) 都 Export Connectors (*.cdk)	

### Figure 4.11. Connectors Context Menu

You	can	show/hide	the	connector	types	by	toggling	the	action
<b>*</b>									

in the toolbar.



### Figure 4.12. Show Connector Types

# 4.5. Properties View

The **Properties View** provides editing capabilities for the currently selected object in Teiid Designer . The selection provided by whichever view or editor is currently in focus will determine the its contents.

To edit a property, click a cell in the Value column. As in the **Table Editor**, each cell provides a UI editor specific to the property type. See Editing Table Properties for a detailed summary.

P	roperties × Description	1: → □
Prop	perty	Value 🔨
в	Info	
	Object URI	
	Searchability	E ALL_EXCEPT_LIKE
	Selectable	🙀 true
	Updateable	Link true
Β	Misc	
	Distinct Value Count	Liii -1
	Name	CATEGORYID
	Name In Source	CATEGORYID
	Null Value Count	L11 -1
8	Туре	
	Datatype	Iong (Path=http://www.w3.org/2001/×
	Native Type	A NUMBER
	Nullable	E NO_NULLS
Β	Type Detail	
	Auto Incremented	😽 false
	Case Sensitive	😽 false
	Character Set Name	LA
	Collation Name	LA
	Currency	🔤 false
	Default Value	LA
<		

#### Figure 4.13. Properties View

If the model for the object being edited is not open in an editor, a dialog may appear confirming the attempt to modify the model and asking the user to confirm or cancel. This dialog can be prevented by checking the preference "Always open editor without prompting". You can re-set/ uncheck this property via the Teiid Designer's main preference page.



### Figure 4.14. Open Model Editor Dialog

Properties can also be edited via a right-click menu presented below.

nt	L1.1 -1		
	LA SHIPNA	ACT.	1
	<b>LA</b> SHIPN/	Undo	
	L11 -1	Cut	
	string (	Сору	
		Paste	۰v.
	LA VARCH	Delete	
	E NULLAI _	Delete	-
		Select All	
	😽 false 👘		_

### Figure 4.15. Open Model Editor Dialog

The **Properties** toolbar contains the following actions:

#### • 🗄

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).

# 4.6. Description View

The **Description View** provides a means to display and edit (add, change or remove) a description for any model or model object. To show the Description View click " **Views > Show View > Other...** " to display the **Show View** dialog. Choose " **Teiid Designer > Description** " view and hit **OK**.



#### Figure 4.16. Description View

You can click in the text field and edit in a similar fashion with standard text editing. Below is a list of available right-click edit actions.

Properties 🖹 Description 🗙	- 8
Northwind_Oracle was created via importi northwin <mark>d schema and Oracle AN</mark> 5I JDBC It is intended to provide basic relational so Designer tool development.	ng from an Oracle database using the connector information. Cut Copy Paste Select All Select All Redo

#### Figure 4.17. Description View Context Menu

# 4.7. Editors

**Editors** are the UI components designed to assist editing your models and to maintain the state for a given model or resource in your workspace. When editing a model, the model will be opened in a **Model Editor**. Editing a property value, for instance, will require an open editor prior to actually changing the property.

Any number of editors can be open at once, but only one can be active at a time. The main menu bar and toolbar for Teiid Designer may contain operations that are applicable to the active editor (and removed when editor becomes inactive).

Tabs in the editor area indicate the names of models that are currently open for editing. An asterisk (\*) indicates that an editor has unsaved changes.

🕞 Northwind_Oracle.xmi	CustomerAccounts.vdb	Rorthwind_View.xmi
<u>هم</u>		TH ADDRESS I

#### Figure 4.18. Editor Tabs

By default, editors are stacked in the editors area, but you can choose to tile them vertically, and or horizontally in order to view multiple models simultaneously.



#### Figure 4.19. Viewing Multiple Editors

The Teiid Designer provides main editor views for XMI models and VDBs.

The Model Editor contains sub-editors which provide different views of the data or parts of data within a model. These sub-editors, specific to model types are listed below.

• Diagram Editor - All models except XML Schema models.

- Table Editor All models.
- Simple Datatypes Editor XML Schema models only.
- Semantic Editor XML Schema models only.
- Source Editor XML Schema models only.

The VDB Editor contains sub-editors which provide different views of the data or parts of data within a VDB. These sub-editors include.

- Overview View and manage contents of VDB.
- **Problems** Display errors and warnings within VDB and models.
- Indexes View indexes generated for model data within VDB.
- User Files Manage additional VDB resources or artifacts.
- Web Services Define and view WSDL.
- Execute Create and change connectors and execute queries.

In addition to general Editors for models, there are detailed editors designed for editing specific model object types. These "object" editors include:

- **Transformation Editor** Manages Transformation SQL for Relational View Base Tables, Procedures and XML Service View Operations.
- Choice Editor Manages properties and criteria for XML choice elements in XML Document View models.
- Input Editor Manages Input Set parameters used between Mapping Classes in XML Document View models.
- Recursion Editor Manages recursion properties for recursive XML Elements in XML Document View models.
- Operation Editor Manages SQL and Input Variables for Web Service Operations.

# 4.8. Problems View

The **Problems View** displays validation errors, warnings, or information associated with a resource contained in open projects within your workspace.

rrors, 12 warnings, 0 infos				
Description	Resource	In Folder	Location	
The VDB model 'Northwind_Oracle.x	CustomerAccounts.vdb	AllCustomerAccounts		
The VirtualDatabase 'CustomerAcco	CustomerAccounts.vdb	AllCustomerAccounts		
The VirtualDatabase 'BooksWebServi	BooksWebService_VDB.vdb	Books Project		
The VirtualDatabase 'BooksXML_VDB'	BooksXML_VDB.vdb	Books Project		
The XML Schema component type fo	BooksXML.×mi	Books Project	bookCollectionDocument/bookCollection/sequ	
The XML Schema component type fo	BooksXML.xmi	Books Project	bookCollectionDocument/bookCollection/sequ	
The XML Schema component type fo	BooksXML.xmi	Books Project	bookCollectionDocument/bookCollection/sequ	
The XML Schema component type fo	BooksXML.xmi	Books Project	bookListingDocument/bookListing/sequence/b	
The XML Schema component type fo	BooksXML.×mi	Books Project	bookListingDocument/bookListing/sequence/b	
The XML Schema component type fo	BooksXML.xmi	Books Project	bookListingDocument/bookListing/sequence/b	
The XML Schema component type fo	BooksXML.xmi	Books Project	bookListingDocument/bookListing/sequence/b	
The XML Schema component type fo	BooksXML.xmi	Books Project	bookListingDocument/bookListing/sequence/b	

### Figure 4.20. Problems View

By default, the **Problems View** is included in the Teiid Designer perspective. If the **Problems View** is not showing in the current perspective click **Views > Show View > Other > Teiid Designer > Problems**.

There are 5 columns:

1. Severity	-	Indicates	the	severity	of	the	line	item	(i.e.,	error
8										,
warning										
▲										,
or										info
i										
).										

- 2. Description A description of the line item.
- 3. **Resource** The name of the resource.
- 4. In Folder The project name.
- 5. Location Model object within the resource that has a validation error.

### 4.8.1. Toolbar Items

Click	the	upside-down	triangle
$\nabla$			
icon	to open the view menu.		



### Figure 4.21. Problems View Drop-down Menu

The **Sorting** action presents a dialog which enables sorting of the problems by severity, resource, and location.



### Figure 4.22. Problems View Sorting Dialog

The **Filters** action presents a dialog which enables filtering of the problems by total number, type, resource, severity, and description.

🙅 Filters	$\overline{\mathbf{X}}$
<ul> <li>Enabled</li> <li>Limit visible items to: 25</li> <li>Show items of type:</li> </ul>	50
Туре	Parent types
Build Path Problem	Problem: Text
EMF Problem	Problem
🗹 Java Problem	Problem; Text
Problem	Marker
VDB Execution	Problem
XML Schema Problem	Problem
<ul> <li>On any resource</li> <li>On any resource in same</li> <li>On selected resource only</li> <li>On selected resource and</li> <li>On working set: <no <no="" li="" set:="" working="" working<=""> </no></li></ul>	project Y I its children rking set selected>
Where description contains	<b>I</b>
Where severity is:	🔲 Error 🔲 Warning 🔲 Info
	Restore Defaults
	OK Cancel

# Figure 4.23. Problems View Filters Dialog

### 4.8.2. Context Menu

Additional actions are available by selecting a problem and right-click to open a context menu.

CustomerAccou
CustomerAccou
in the Million Million
В.
Navigator
All I
x
ies

### Figure 4.24. Problems View Context Menu

- Go To will open the appropriate editor and select the affected/referenced object.
- Show In Navigator Opens the Basic > Navigator view (if not open) and expands file system tree and reveals applicable resource.
- Copy Copies the problem information to the system clipboard.
- **Paste** Pastes the problem information located in the system clipboard ( if applicable ) into the curor location for a specified text editor.
- Delete Deletes the selected problem rows ( if applicable ).
- Select All selects all problems in the table.
- Quick Fix (Not yet implemented in Teiid Designer).
- Properties displays a dialog containing additional information.

# 4.9. Message Log View

The **Message Log** view provides a console-based view of any errors or warnings that may occur during your **Teiid Designer** session.

System Catalog Problems 🥵 Message Log 🗙		0,   0, × 0 6 × - 0
Message	Plug-in	Date
🥺 Error processing query	net.sourceforge.sqlexplorer	2007-08-15 11:21:26.597

#### Figure 4.25. Message Log View

You can perform various operations on the content of the viewer via the toolbar actions or use the right-click menu shown below.



#### Figure 4.26. Message Log View Context Menu

# 4.10. Preview Results View

Executing	the	Preview	Data	action
R.				

, will result in the query results being displayed in the Preview results view, shown below.

roblems Message Log 🔂 Preview 🗙					20 of 20 rows	- E	i 🎋 🕴	• 🗊 🚰 🗖
History	AUTHOR_ID	FIRSTNAME	LASTNAME	MIDDLEINIT				~
ALITHORS - (TestParts/Books, O. 9), C.xmi	2	Jeff	Rashka					
Contract ALITHOPS - (TestParts/Books, O. 9) WS yrold	3	John	Paul					
getAUTHORS - / TestParts/Books_U_9I_wS.xml	4	David	Marco					
	5	Alfred	Aho	۷.				
	6	Ravi	Sethi					
	7	Jeffrey	Ullman	D.				
	8	Ron	Black					
	9	Magee	Kramer					
	10	Douglas	Lea					
	11	Karl	Wiegers	Ε.				
	12	Robert	Muller	J.				
	13	Hector	Garcia-Molina					
< >	14	Fergus	O'Connell					~

### Figure 4.27. Preview Results View

The view's toolbar actions include:

Show Query Results - Shows query results in table row or XML format.

#### • E

Show Query Plan Tree - Shows query plan in tree format.

• 🗄

Show Query Plan Document - Shows query plan in xml document format.

#### • 🏠

Show Debug Log - Shows debug log file contents.

• 🗙

Close the Selected Results - Removes the selected results from the view.

Copy Results to Clipboard - Copies selected results as text to the system clipboard.

# • 🚘

Save Results to File - Provides a file dialog to save selected results to a text file on file system.

# Tip

The History list can be sorted alphabetically (or reverse) by clicking the "History" column header.



Because the **Preview Data** feature is just that, a preview, you are limited to maximum number of rows returned rows. This value is a user preference and can be changed from its default value by selecting **Preferences > Teiid Designer** and changing the **Preview Data's** row count limit value.

By selecting the **Show Query Plan Tree** toolbar action the SQL table results are replaced with a tree view of the underlying Query plan and corresponding properties.

Problems Message Log 🕞 Preview 🗙	20 of 20 rows 📰 🙀 🌼 🙀 🖶 🗎 🖓
History AUTHORS - /TestParts/Books_O_9i_C.xmi GetAUTHORS - /TestParts/Books_O_9i_W5.xmi/	Plan Nodes Project [Books_O_9i_C.AUTHORS.AUTHOR_ID, Books_O_9i_C.AUTHORS.FIRSTNAME, Books_O_9i_C.AUTHORS.LASTNA Access [SELECT Books_O_9i_C.AUTHORS.AUTHOR_ID, Books_O_9i_C.AUTHORS.FIRSTNAME, Books_O_9i_C.AUTHORS.AUTHOR_ID, Books_O_9i_C.AUTHORS.FIRSTNAME, Books_O_9i_C.AUTHORS.AUTHOR_ID, Books_O_9i_C.AUTHORS.FIRSTNAME, Books_O_9i_C.AUTHORS.AUTHOR_ID, Books_O_9i_C.AUTHORS.FIRSTNAME, Books_O_9i_C.AUTHORS.AUTHOR_ID, Books_O_9i_C.AUTHORS.FIRSTNAME, BOOKS_O_9i_C.FIRSTNAME, BOOKS_O_9i_C.FIRSTNAME, BOOKS_O_9i_C.FI
	Properties Output Columns AUTHOR_JD (long) FIRSTNAME (string) STMAME (string)
×>	

#### Figure 4.28. Search Results View

By selecting the **Show Query Plan Document** toolbar action the SQL table results are replaced with an XML document view of the underlying Query plan.



Figure 4.29. Search Results View

# 4.11. Search Results View

Below is an example set of search results. The view contains rows representing matches for your search parameters. You can double-click a entry and the object will be opened and selected in an editor and/or the VDB Explorer if applicable.

Problems	Message Log	Progress	Repository Console	🖋 Classic Search 🗙		Ŷ	Ŷ	× %	1	9 🗐	
Found 66	Model Objects										
e audio	Book : /Books/	Books.xsd/	schema/BookSetMixe	d/choice/audioBook - http:/	/www.eclipse.org/xsd/2002/>	(SD#/)	/XSDE	lement	Declara	ation	^
🔚 Audio	Book : /Books/	Books.xsd/	/schema - http://www	eclipse.org/xsd/2002/XSD.	#//XSDComplexTypeDefinition	n					
🗾 BOOI	<_AUTHORS : /	Books/Bool	ks_Oracle.xmi/BOOKS	- http://www.metamatrix.		//Base	Table				
BOO	-AUTHORS : /	Books/Bool	ks_SQL.xmi - http://w	ww.metamatrix.com/metar	nodels/Relational#//BaseTabl	е					
e book	: /Books/Books	.xsd/scher	na/BookSetFlat/seque	ence/book - http://www.ecl	pse.org/xsd/2002/XSD#//XSI	DEleme	entDe	claratio	n		
e book	: /Books/Books	.xsd/scher	na/BookSetNested/se	quence/book - http://www	eclipse.org/xsd/2002/XSD#//	XSDE	emen	Declara	ation		
e book	Collection : /Bo	oks/Books.	xsd/schema - http://v	/www.eclipse.org/xsd/2002/	XSD#//XSDElementDeclaration	п					
S Book	Datatypes : /Bo	ooks/BookD	atatypes.xsd - http:/	//www.eclipse.org/xsd/200	2/XSD#//XSDSchema						
Book	Edition : /Books	/BookData	types.xsd/schema - h	http://www.eclipse.org/xsd,	2002/XSD#//XSDSimpleTypel	Definiti	ion				
Book	Edition : /Books	/BookData	types.xsd/schema - h	http://www.eclipse.org/xsd	2002/XSD#//XSDSimpleTypel	Definit	ion				
🖪 Book	Flat : /Books/Bo	ooks.xsd/so	thema - http://www.e	eclipse.org/xsd/2002/XSD#,	//XSDComplexTypeDefinition						
e book	Listing : /Books	/Books.xsd	//schema - http://www	v.eclipse.org/xsd/2002/XSE	#//XSDElementDeclaration						
Book	Nested : (Book	s/Books.xsi	dischema - http://ww	w.eclinse.ora/xsd/2002/XS	D#1/XSDComplexTypeDefiniti	nn					×
<											>

### Figure 4.30. Search Results View

The toolbar actions for the Search Results view are:

### • 🕂

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.

#### S S

Search - Launches the MoTeiid Designerearch Dialog.

#### • 🍕

Previous Search Results - Select previous search results from history.

You can also perform some of these actions via the right-click menu:

Problems	Message Log	Progress	Repository Console	🖋 Classic Search	×
Found 66	Model Objects				
e audiol 🖫 Audio	Book : /Books/B Book : /Books/B	Books.xsd/ Books.xsd/	/schema/BookSetMixe /schema - http://www	d/choice/audioBook v.eclipse.org/xsd/20	- http://w 02/XSD#/
BOOK	_authors : /f _authors : /f	Books/Bo Books/Bo	🔏 Go to File	· [ 1] · · ·	n i co mo
e book e book	: /Books/Books : /Books/Books	.xsd/schi .xsd/schi -	Copy to Clipboard	ļ	lips 
e book( S BookE	Collection : /Boo Datatypes : /Bo Edition : /Books.	oks/Book oks/Booł /BookDat	🗶 Remove Selected 🔆 Remove All Match	Match es	(XS  2/)  /2(
BookE	dition : /Books, lat : /Books/Bo	/BookDat oks.xsd/	Compare With Replace With		<ul> <li>► 1/20</li> <li>► 1/X</li> </ul>
	listing : (Books) Jested : (Books	BOOKS.XS i/Books.X	Search Again		ע#, יר#
<			🗞 Add Selection to N 🏠 Add All to Metada	1etadata Favorites ta Favorites	J

# Figure 4.31. Search Results Context Menu

# 4.12. Datatype Hierarchy View

To open **Teiid Designer's Datatype Hierarchy** view, select the main menu's **Views > Show View > Other...** and select the **Teiid Designer > Datatypes** view in the dialog.

Model Explorer Outline 😫 Datatype Hierarchy 🗙	- 0
🖃 🗖 anySimpleType : xs:anyType	^
🗖 anyURI	
主 🗖 base64Binary	
🗖 boolean	
🗖 date	
<b>D</b> dateTime	
连 🗖 decimal	
<b>D</b> double	
uration	=
🖬 float	-
🔤 🖬 gDay	
🗖 gMonth	
🔤 🗖 gMonthDay	
<b>D</b> gYear	
gYearMonth	
hexBinary	
- QName	
E string	
🗖 char : xs:string	
Clob : xs:string	*
Datatype details	
Property Value Defined by	
whiteSpace preserve string length 1	

### Figure 4.32. Datatype Hierarchy View

# 4.13. Model Classes View

The Model Classes View....

# 4.14. System Catalog View

To open **Teiid Designer's System Catalog** view, select the main menu's **Views > Show View > Other...** and select the **Teiid Designer > System Catalog** view in the dialog..



### Figure 4.33. System Catalog View

# 4.15. Database Structure View

The Database Structure View....

🗄 Databas	Database Structure View 🛛 🗖 🗖								
PartsSuppli	PartsSupplier_VDB.vdb-13:34:49 Aug 03								
<ul> <li>➡ Database</li> <li>➡ PartsSupplier_VDB</li> <li>➡ PartsVirtual.partsByColor</li> <li>➡ PartsVirtual.partsByColor</li> <li>➡ System</li> <li>➡ Table</li> <li>➡ PartsSupplier_SourceA.PARTS</li> <li>➡ PartsSupplier_SourceA.SHIP_VIA</li> <li>➡ PartsSupplier_SourceA.SUPPLIER</li> <li>➡ PartSupplier_SourceB.SUPPLIER_PARTS</li> <li>➡ PartSupplier_SourceB.SHIP_VIA</li> <li>➡ PartSupplier_SourceB.SHIP_VIA</li> <li>➡ PartSupplier_SourceB.SUPPLIER_PARTS</li> <li>➡ PartSupplier_SourceB.SUPPLIER_PARTS</li> <li>➡ PartSupplier_SourceB.SUPPLIER</li> </ul>									
Columns	Indexes	Primary K	ey Forei	gn Key 🗎	Preview F	Row Count			
Colu	Data	Size	Deci	Defa	Acce	Com			
SHIPP	short	2	0		YES				
SHIPP short 2 0 YES SHIPP string 30 0 NO									
1									

### Figure 4.34. Database Structure View

# 4.16. SQL History View

The SQL History View....



### Figure 4.35. SQL History View

# 4.17. Connections View

The Connections View....



### Figure 4.36. Connections View

# 4.18. SQL Editor View

The SQL Editor



### Figure 4.37. SQL Editor View

# 4.19. SQL Results View

The SQL Results View....

lect ISBN, BooksWebSe	TITLE, SUBTITLE, PUBLISHER, PUBLISH_YEAR, ED: rvice_VDB"."Books_SourceB.BOOKS"	ITION, TYPE from
SQL Results 🗴	41 of 41 rows	🗖 🗄 🖟 🛸 🛤 🚰 '
1		
ISBN	TITLE	SUBTITLE
ISBN	TITLE After the Gold Rush	SUBTITLE
ISBN )-7356-0877-6 )-201-43287-0	TITLE After the Gold Rush Automated Software Testing	SUBTITLE Creating a True Profession of Soft Introduction, Management, and P
ISBN 0-7356-0877-6 0-201-43287-0 1-55615-484-4	TITLE After the Gold Rush Automated Software Testing Code Complete	SUBTITLE Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software
ISBN 0-7356-0877-6 0-201-43287-0 1-55615-484-4 0-201-10088-6	TITLE After the Gold Rush Automated Software Testing Code Complete Compilers	SUBTITLE Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles. Techniques, and Tools
I5BN -7356-0877-6 -201-43287-0 1-55615-484-4 -201-10088-6 -7897-2271-2	TITLE After the Gold Rush Automated Software Testing Code Complete Compilers Compilers Complete Idiot's Guide to Project Management with Microsoft Project 2000	SUBTITLE Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools
ISBN 1-7356-0877-6 0-201-43287-0 1-55615-484-4 0-201-10088-6 0-7897-2271-2 0-471-98710-7	TITLE After the Gold Rush Automated Software Testing Code Complete Compilers Complete Idiot's Guide to Project Management with Microsoft Project 2000 Concurrency State Models & Java Programs	SUBTITLE Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools
ISBN 1-7356-0877-6 0-201-43287-0 1-55615-484-4 0-201-10088-6 1-7897-2271-2 1-471-98710-7 0-201-31009-0	TITLE After the Gold Rush Automated Software Testing Code Compilets Compilers Compilete Idiot's Guide to Project Management with Microsoft Project 2000 Concurrency State Models & Java Programs Concurrent Programming in Java	SUBTITLE Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns
ISBN -7356-0877-6 0-201-43287-0 1-55615-484-4 0-201-10088-6 0-7897-2271-2 0-471-98710-7 0-201-31009-0 0-932633-33-1	TITLE After the Gold Rush Automated Software Testing Code Complete Compilers Complete Idiot's Guide to Project Management with Microsoft Project 2000 Concurrency State Models & Java Programs Concurrent Programming in Java Creating a Software Engineering Culture	SUBTITLE Creating a True Profession of Sofi Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns
ISBN D-7356-0877-6 D-201-43287-0 1-55615-484-4 D-201-10088-6 D-7897-2271-2 D-471-98710-7 D-201-31009-0 D-932633-33-1 1558605150	TITLE         After the Gold Rush         Automated Software Testing         Code Complete         Compilers         Complete Idiot's Guide to Project Management with Microsoft Project 2000         Concurrency State Models & Java Programs         Concurrent Programming in Java         Creating a Software Engineering Culture         Database Design for Smarties	SUBTITLE Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns Using UML for Data Modeling
ISBN 0-7356-0877-6 0-201-43287-0 1-55615-484-4 0-201-10088-6 0-7897-2271-2 0-471-98710-7 0-201-310090 0-932633-33-1 1558605150 0-13-239856-7	TITLE         After the Gold Rush         Automated Software Testing         Code Complete         Compilers         Complete Idiot's Guide to Project Management with Microsoft Project 2000         Concurrency State Models & Java Programs         Concurrent Programming in Java         Creating a Software Engineering Culture         Database Design for Smarties         How to Run Successful Projects II	SUBTITLE Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns Using UML for Data Modeling
ISBN 0-7356-0877-6 0-201-43287-0 1-55615-484-4 0-201-10088-6 0-7897-2271-2 0-471-98710-7 0-201-31009-0 0-932633-33-1 1558605150 0-13-239856-7 0-201-31000-7	TITLE         After the Gold Rush         Automated Software Testing         Code Complete         Compilers         Concurrency State Models & Java Programs         Concurrent Programming in Java         Creating a Software Engineering Culture         Database Design for Smarties         How to Run Successful Projects II         Inside Java 2 Security	SUBTITLE Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns Using UML for Data Modeling

Figure 4.38. SQL Results View

# **Editors**

#### Go To: Table of Contents [index.html]

**Editors** are the UI components designed to assist editing your models and to maintain the state for a given model or resource in your workspace. When editing a model, the model will be opened in a **Model Editor**. Editing a property value, for instance, will require an open editor prior to actually changing the property.

Any number of editors can be open at once, but only one can be active at a time. The main menu bar and toolbar for Teiid Designer may contain operations that are applicable to the active editor (and removed when editor becomes inactive).

Tabs in the editor area indicate the names of models that are currently open for editing. An asterisk (\*) indicates that an editor has unsaved changes.

🕞 Northwind_Oracle.xmi	CustomerAccounts.vdb	👫 Northwind_View.xmi 🗙
(A)		Planetara I

### Figure 5.1. Editor Tabs

By default, editors are stacked in the editors area, but you can choose to tile them vertically, and or horizontally in order to view multiple models simultaneously.

👫 Northwind_Oracle.xmi 🗙 📃 🗖	Rorthwind_View.xmi	8
Image: Customer Tables >>         Image: Customer Tables Tab	Image: Section of the section of th	
CustomerArcounts with 8		
Last Updated: Aug 15, 2007 11:20:57 AM		^
Status: 👍 WARNING		
Validation: 🔽 Validate for MetaMatrix Server / DQP Deployment		
Models:		
Index     I	Add	
AllCustomerAccounts/Northwind Aug 15, 20 Synchronized	Remove	
	Synchronize	-
	Synchronize All	
Quarview Drohlemr Indexed Web Services Evente	Edit	~

Figure 5.2. Viewing Multiple Editors

Editors also contain sub-editors which provide different views of the data or parts of data within a model. These sub-editors, specific to model types are listed below.

- Diagram Editor All models except XML Schema models.
- Table Editor All models.
- Simple Datatypes Editor XML Schema models only.
- Semantic Editor XML Schema models only.
- Source Editor XML Schema models only.

In addition to general Editors for models, there are detailed editors designed for editing specific model object types. These "object" editors include:

- <u>Transformation Editor</u> Manages Transformation SQL for Relational View Base Tables, Procedures and XML Service View Operations.
- <u>Choice Editor</u> Manages properties and criteria for XML choice elements in XML Document View models.
- <u>Input Editor</u> Manages Input Set parameters used between Mapping Classes in XML Document View models.
- <u>Recursion Editor</u> Manages recursion properties for recursive XML Elements in XML Document View models.
- Operation Editor Manages SQL and Input Variables for Web Service Operations.

### 5.1. Model Editor

The Model Editor is comprised of sub-editors which provide multiple views of your data. The Diagram Editor provides a graphical while the Table Editor provides spreadsheet-like editing capabilities. This section describes these various sub-editors.

### 5.1.1. Diagram Editor

The Diagram Editor provides a graphical view of the a set of model components and their relationships.

Several types of diagrams are available depending on model type. They include:

• 點

Package Diagram

```
• •
```

Custom Diagram

• ⊷ჭ

**Transformation Diagram** 

• Rapping Diagram

• ⊷€

**Mapping Transformation Diagram** 

• • • • •

#### **Relationship Diagram**

You can customize various diagram visual properties via Diagram Preferences.

Each diagram provides actions via the Main toolbar, diagram toolbar and selection-based context menus. These actions will be discussed below in detail for each diagram type.

When a **Diagram Editor** is in focus, a set of common diagram actions is added to the application's main toolbar.

```
🖻 100% 💽 👂 🗛 💒 🔡
```

### Figure 5.3. Main Toolbar Diagram Actions

- The actions include:
  - 👂

Zoom In



Zoom to Level

P Zoom Out

A Increase Font Size

A Decrease Font Size

Perform Diagram Layout

### 5.1.1.1. Package Diagram

The Package Diagram provides a graphical view of the contents of a model container, be it the model itself, a relational catalog or schema.



### Figure 5.4. Package Diagram Example

- Package Diagram toolbar actions include:
  - 🏟

Refresh Diagram - Re-draws diagram.

• 🔁

Show Parent Diagram - Navigates to diagram for parent object (if available).

• 록

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...

Ø

Show/Hide Page Grid - Show current page boundaries as grid in diagram.

Context menus provide a flexible means to edit model data, especially from Package Diagrams. Each Package Diagram represents the contents of some container (i.e. Model, Category, Schema, etc...), so New Child, New Sibling and New Association actions are almost always available in addition to standard Edit actions (Delete, Cut, Copy, Paste, Rename, Clone).

A sample context menu for a relational base table is shown below.

New Child	: <b>•</b>
New Sibling	•
New Association	•
Modeling	•
💛 Undo Set org.eclifalse) (iD: false) to	o 5.5
😂 Redo	
o∉ Cut	
Сору	
🖹 Paste	
Clone	
Copy Name	۲
💢 Delete	
Rename	
Open	Ctrl+O
Edit	
🗞 Add to Metadata Favorites	
🚯 Open in Relationship Navigator	
Link Types	۲
💖 Refresh Diagram	
🔁 Show Parent Diagram	
🗟 Save Diagram To File	
ි <sup>න</sup> Page Setup	
🕌 Show Page Grid	

### Figure 5.5. Package Diagram Contenxt Menu

### 5.1.1.2. Custom Diagram

The **Custom Diagram** represents a view of user-defined model objects. Unlike **Package Diagrams**, **Custom Diagrams** can contain objects that are not only unrelated, but can be from different containers and even models.


# Figure 5.6. Package Diagram Example

- Custom Diagram toolbar actions include:
  - ଂ 🏟

Refresh Diagram - Re-draws 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.

Since **Custom Diagrams** do not represent represents the contents of container objects (i.e. Model, Category, Schema, etc...) its **context menus** are limited to adding/removing objects from diagram and basic diagram-related display options.

# 5.1.1.3. Transformation Diagram

The **Transformation Diagram** represents a view of the relationships defined by the source inputs described in a view table's SQL transformation.

🔞 Parts	sVirtual.xmi 🗙 🕞 PartSupplier_SourceB.xm		- 0
	SVirtual.xmi X PartSupplier_Source8.xm	ccBase Table>>         SUPPLIER_PARTS         PartSupplier_SourceB         SUPPLIER_ID : string(10)         PART_ID : string(4)         AutomITY : short         SHIPPER_ID : short         FK_SPLIER_PRTS_PRTS         FK_SPLY_PRTS_SPLY         PK_SUPPLIER_PARTS         PK_SUPPLIER_PARTS	
Rái Trai	orformation Discram 🥅 Table Editor		>

# Figure 5.7. Transformation Diagram Example

- Transformation Diagram toolbar actions include:
  - 🏟

Refresh Diagram - Re-draws diagram.

1

Show Parent Diagram - Navigates to diagram for parent object (if available).

• 록

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...

•

Ø

Show/Hide Page Grid - Show current page boundaries as grid in diagram.

Context menus for the

#### 5.1.1.4. Mapping Diagram

The **Mapping Diagram** represents a view of the mapping between virtual mapping class columns and XML document elements. This mapping defines how source data is transformed from row-based results into XML formatted text.

💦 BooksXML.xmi 🗙						
Image: Sequence [MC: book]         Im	««Mapping Class»»     E book     A isbn : string     A subtitle : string     A edition : string     A publishDate : string     A publishDate : string     author     A author     A author : string     A author     A author : string     A author : string     A author     A author : string     A author     Author     A Author     A Author     A Auth					
Hig Mapping Diagram						

# Figure 5.8. Mapping Diagram Example

- Mapping Diagram toolbar actions include:
  - 🏟

Refresh Diagram - Re-draws 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.

• 록

Preview Data - Executes a simple preview query (SELECT \* FROM ).

# • \*₽

**Generate Mapping Classes** - Generate mapping classes for the selected XML document root element.

# ' \*<mark>=</mark>

**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 Classe - Merge selected mapping classes.

• •

Split Mapping Class - Split selected mapping class.

• 🗗

**Display All Mapping Classes** 

• **E** 

Show Mapping Class Columns

• ∔

Filter Displayed Mapping Classes with Selection

**Context menus** for **Mapping Diagrams** provide Edit capability to the mapping class in addition to mapping class manipulation actions (i.e. Merge Mapping Classes, Split Mapping Class, etc..)

# 5.1.1.5. Mapping Transformation Diagram

The **Mapping Transformation Diagram** is identical to a Transformation Diagram except for displaying an Input Set and possibily Staging Tables as sources for the Mapping Class's transformation.

BooksWitz.com × bookCollecton } bookCollecton } concerned book	VIEW	SOURCES		
	<pre>cotapping Class&gt;</pre>	<choput set="">&gt; ■ Input Set</choput>	ediase Tablese     Bolds_Source     District Streng(255)     SUBTILE: string(255)     SUBTILE: string(255)     SUBTILE: string(255)     DICLISHER: Int     DPUBLISHER: Int     DPUBL	
👎 Diagram 🔲 Table Editor				

#### Figure 5.9. Mapping Transformation Diagram Example

- Mapping Transformation Diagram toolbar actions include:
  - ଂ 🏟

Refresh Diagram - Re-draws diagram.

• 💼

Show Parent Diagram - Navigates to diagram for parent object (if available).

• 록

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...

**Context menus** for **Mapping Transformation Diagrams** identical capabilities to the Transformation Diagram with the addition of managing and editing Input Sets.

#### 5.1.1.6. Relationship Diagram

The **Relationship Diagram** represents a view of relationships between sources and targets defined in your Relationship models as well as the relationships and hierarchy between relationship types.



#### Figure 5.10. Relationship Diagram Example

- Relationship Diagram toolbar actions include:
  - 🏟

Refresh Diagram - Re-draws diagram.

#### ່ 🔁

Show Parent Diagram - Navigates to diagram for parent object (if available).

# • 🚽

Remove From Relationship - Removes the selected source or targets from a relationship.

• 🖫

Save Diagram as Image - Save the diagram image to file in JPG or BMP format.

# 5.1.2. Table Editor

The **Table Editor** provides a table-based object type structured view of the contents of a model. The figure below shows a relational model viewed in the **Table Editor**. Common object types are displayed in individual folders/tables. All base tables, for instance, are shown in one table independent of their parentage.

ation	Name	Name In So	System	Cardinality	Supports U	Materialized	Logical Rela	Object URI	Description
	CATEGORIES	CATEGORIES	false	7	true	false			
	CUSTOMER	CUSTOMER	false	0	true	false			
	CUSTOMER	CUSTOMER	false	0	true	false			
	CUSTOMERS	CUSTOMERS	false	91	true	false			
	EMPLOYEES	EMPLOYEES	false	8	true	false			
	EMPLOYEE	EMPLOYEE	false	42	true	false			
	ORDERS	ORDERS	false	787	true	false			
	ORDER_DE	ORDER_DE	false	1737	true	false			
	PRODUCTS	PRODUCTS	false	65	true	false			
	REGION	REGION	false	4	true	false			
	SHIPPERS	SHIPPERS	false	3	true	false			
	SUPPLIERS	SUPPLIERS	false	29	true	false			
	TERRITORIES	TERRITORIES	false	53	true	false			

#### Figure 5.11. Table Editor Example

You can customize Table Editor properties via Table Editor Preferences.

These are the primary features of the Table Editor:

- Edit existing properties.
- Add, remove or edit objects, via the main Edit menu and context menu (Cut, Copy, Paste, Clone, Delete, Rename, Insert Rows).

- Paste information from your clipboard into the table.
- Print your tables.

When a **Table Editor** is in focus, the **Insert Table Rows** action

is added to the application's main toolbar.

A few Table Editor actions are contributed to the right-click menu for selected table rows. These actions, described and shown below include:

- Table Paste Paste common spreadsheet data (like Microsoft Excel) to set object properties.
- **Table Editor Preferences** Change table editor preferences, including customizing visible properties.
- Insert Rows Create multiple new sibling objects.
- **Refresh Table** Refreshes the contents of the current Table Editor to insure it is in sync with the model.



Figure 5.12. Table Editor Example

# 5.1.2.1. Editing Properties

You can edit properties for an object by double-clicking a table cell.

For String properties, the table cell will become an in-place text editor field.

Base	Tables 🛛 🖾 Columns	E Foreign	
ation	Name	Na	
	CATEGORIES	C/	
	CUSTOMERCUSTOMERDEMO		
	CUSTOMERDEMOGRAP	HICS CL	
	CUSTOMERS	CL	
	EMPLOYEES		

Figure 5.13. Editing String Property

If a property is of a boolean (true or false) type or has multiple, selectable values, a combo box will be displayed to change the value.

Support	s U
true	
true	-
true	
Frite	

Figure 5.14. Editing Boolean Value

	Searchability	Currenc	
	ALL_EXCEP	false	
	ALL_EXCEP	false	
	ALL_EXCEP	false	
	ARCHABLE -	false	
SEARCHABLE			
ALL_EXCEPT_LIKE			
	LIKE_ONLY		
	UNSEARCHABLE		
SEARCHABLE false			
	SEARCHABLE	false	

Figure 5.15. Editing Multi-Value Property

For multi-valued properties where the available values are dynamic (i.e. can change based on available models or data), a picker-button ("....") will be displayed.

Datatype	1
int : xs:long	
short : xs:int	
long : xs:integer	
string (Path=http://www.w3.org/200	
long : xs:integer	
string	
string	
long : xs:integer	
chripa	

#### Figure 5.16. Editing Multi-Value With Picker

An example of of this type is the relational column datatype property. Editing via the table cell and clicking the "..." button for datatype will display the following dialog.

🙅 Select a Datatype	×
Select a Datatype for this Column (Only types with Enterprise information are shown)	
type filter text	Clear
Select datatypes to show	
Built-In Datatypes     Iser-Defined Datatypes	
long : xs:integer	~
Name : xs:token	
NCName : xs:Name	
negativeInteger : xs:nonPositiveInteger	
NMTOKEN : xs:token	
NMTOKENS : xs:NMTOKEN	
nonNegativeInteger : xs:integer	
nonPositiveInteger : xs:integer	=
normalizedString : xs:string	5
NOTATION	
object : xs:base64Binary	
positiveInteger : xs:nonNegativeInteger	-
Details	
 built-in> long	
OK Can	cel

Figure 5.17. Editing Datatype Values

### 5.1.2.2. Inserting Table Rows

The Insert Rows action provides an additional way to create objects in a model. Insert Rows action performs the same function as Insert SIbling action, but allows you to create multiple children at the same time. All new rows will correspond to an object of the same type as the selected object and be located under the same parent as the selected object.

To Insert Rows in a table:

Step 1: Select a table row to insert rows after.

**Step 2:** Right-click select "Insert Rows" action or select the Insert Rows action on the main toolbar. The following dialog will be displayed.

🙅 Insert Table Rows	
Insert Rows (Maximum of 1000) 10 rows will be inserted	
Number of Rows: 10 🚔	OK Cancel

#### Figure 5.18. Editing String Property

**Step 3:** Edit the Number of Rows value in the dialog, or use the up/down buttons to change the value.

Step 4: Select OK in dialog.

The desired number of rows (new model objects) will be added after the original selected table row.

# 5.1.3. Simple Datatypes Editor

The Simple Datatype Editor provides a form-based properties view of XML Schema data.

🔞 Northwind_Oracle.xmi 💦 📴 Boo	okDatatypes.xsd 🗙 🖓 🗖					
New Delete	Simple Datatype Editor					
pe filter text Clear	pe filter text Clear Target Namespace: http://www.metamatrix.com/XMLSchema/DataSets/Books/BookDatatypes (change)					
Vatatypes in tookDatatypes vsd						
Name: ISBN (change)						
Image: Description     Description       ISBN     ISBN, using very simple formatting rules.       PublicationDate     PublicationPare						
	▼ Inheritance					
	Base datatype: <u>string</u> (http://www.w3.org/2001/XMLSchema) ( <u>change)</u>					
	Create subtype of this datatype Prevent Restrictions:  Atomic					
	Open in Datatype Hierarchy view  Union					
	▼ Enterprise					
	Information in this section is required in order to use this type in other models.					
l Hide Built-in Datatypes	Runtime type: string					
Built-in Datatypes:	▼ Length					
anySimpleType anyUR1 base64Binary bigdecimal biginteger biob boolean byte	Length       0       →       Fixed       ✓ Use Default       Add Description         Minimum Length       0       →       ●       Fixed       ✓ Use Default       Add Description         Maximum Length       0       →       ●       Fixed       ✓ Use Default       Add Description         Maximum Length       0       →       ●       Fixed       ✓ Use Default       Add Description					
char 📃						
date						
dateTime decimal double	Enumerations Value Description Add Edit					
ENTITY	Remove					
float gDay aMostb	Patterns Value Description Add Use Default					
gMonthDay	[0-9,\-,X]{10,13}					
gYear 🗸	Remove					
🔲 Simple Datatypes 🥅 Table Edito	S Semantics Source					

# Figure 5.19. Editing String Property

# 5.1.4. Semantic Editor

The Semantic Editor is a tree based editor for XML Schema elements and attributes.

# 5.1.5. Source Editor

The Source Editor is a simple text editor which is aware of XML Schema formatting rules.

# 5.1.6. Model Object Editors

The Model Object Editors represent specialized sub-editors which are available for specific model object types.

- For details, select a specific editor listed below:
  - Transformation Editor

XML Mapping Input Set Editor XML Choice Editor XML Recursion Editor

Web Service Operation Editor

# 5.2. VDB Editor

A **VDB**, or **virtual database** is a container for components used to integrate data from multiple data sources, so that they can be accessed in a federated manner through a single, uniform API. A **VDB** contains models, which define the structural characteristics of data sources, views, and Web services. The **VDB Editor**, provides the means to manage the contents of the **VDB** as well as its deployable (validation) state.

- The VDB Editor, shown below, contains six editor tabs, namely:
  - Overview Manage/edit contents, description and indicates current deployable state.
  - <u>Problems</u> Table view of any validation problems your VDB may have (i.e. Warnings and/ or Errors).
  - <u>Indexes</u> Provides viewable summary of various Index files utilized at run-time.
  - User Files Provides importing or attaching additional user files to the VDB archive.
  - <u>Web Services</u> When VDB contains Web Services models, this tab provides access edit and view WSDL specific information.
  - Execute Manage source model connector bindings and actually test your data.

🔋 PartsVDB.v	db ×			- 0
Name:	PartsVDB			
Description:				<
Last Updated	Cot 8, 2008 4:37:03 PM			<u> </u>
Status:	ок			
Validation:	🔽 Validate for MetaMatrix Server / D	QP Deployment		
_Models: -				
Visibility	Model	Added	Workspace	
	PartsProjectA/Physical_Sources/ PartsProjectA/Views/Parts_V.xmi	Oct 08, 20 Oct 08, 20	Synchronized Synchronized	<u>A</u> dd <u>R</u> emove Synchronize
				Synchronize All
Notes:     Checked (     Changes     To update     All button	models will be visible to clients when this made to models in the Designer workspa e the VDB with changes, select any mode 1.	VDB is deployed. ce are not automa els marked "Not Syr	tically synchronized with this VDB, nchronized" and press the Synchror	ize button or press Sychronize
Overview Pro	blems Indexes User Files Web Service	es Execute		

#### Figure 5.20. VDB Editor

# 5.2.1. Overview

The **Overview** tab allows you to Manage and edit the contents, description and indicates current deployable state of your VDB.

Manage content of the VDB by using the Add... and Remove... buttons.

**Synchronize** and **Synchronize All** button provide a easy way to update the contents of your VDB if changes have occurred to one or more models in your local workspace. Remember, when a model is added to a VDB, an exact copy of that model is added the the VDB archive. If a model has changed since it was added to a VDB, a **Not Synchronized** message will be visible in the Workspace Status column.

An example VDB containing multiple models is shown below (including models that have been modified in the workspace after they were added to the VDB).

🗍 CustomerAc	counts.vdb 🚺 BooksWebService_VDB	.vdb ×		- 8	
Name:	BooksWebService_VDB				
Description: This virtual database (VDB) definition file contains the models required to deploy the Web service of the Books project.					
Last Updated:	Aug 16, 2007 11:04:17 AM				
Status:					
Validation:	Validate for MetaMatrix Server / DQP De	ployment			
_Models:					
Visibility M	odel	Added	Workspace Status		
	Books Project/BookDatatypes.xsd	Apr 07, 2006 13:39:57	Synchronized	Add	
	Books Project/Books_SourceA.xmi	Apr 07, 2006 13:39:57	Not Synchronized		
	Books Project/Books_SourceB.xmi	Apr 07, 2006 13:39:57	🚯 Not Synchronized	Remove	
2	Books Project/Books.xsd	Apr 07, 2006 13:39:57	Synchronized	Synchronize	
	Books Project/BooksInput.xsd	Apr 07, 2006 13:39:57	Synchronized		
🗹 👊	Books Project/BooksWebService.xmi	Jul 25, 2006 13:22:32	🚯 Not Synchronized	Synchronize All	
	Books Project/BooksXML.xmi	Apr 07, 2006 13:39:57 (	1 Not Synchronized	Edit	
Notes:     Otecked m     Changes m     To update     button.	odels will be visible to clients when this VDB is ade to models in the Designer workspace are the VDB with changes, select any models ma lems Indexes Web Services Execute	; deployed. ; not automatically synchronize; rked "Not Synchronized" and pr	d with this VDB. ess the Synchronize button or press	Sychronize All	

#### Figure 5.21. VDB Editor Overview Panel

# 5.2.2. Problems

The **Problems** tab displays all errors and warnings for your VDB and its contained models.

_	Books Project (BooksWebService, VDB vdb	Object
	"isbn" references an XML Schema component that has a min occurs of zero, and may not appear in a result document.	/Books Project/BooksXML.xmi
	"type" references an XML Schema component that has a min occurs of zero, and may not appear in a result document.	/Books Project/BooksXML.xmi
	"author" references an XML Schema component that has a min occurs of zero, and may not appear in a result document.	/Books Project/BooksXML.xmi
	"isbn" references an XML Schema component that has a min occurs of zero, and may not appear in a result document.	/Books Project/BooksXML.xmi

#### Figure 5.22. VDB Editor Problems Panel

#### 5.2.3. Indexes

The **Indexes** tab provides visual access to the underlying metadata that will be deployed to the server and accessed at runtime.

🚺 Cust	tomerAccounts.vdb	BooksWebService_VDE	VDB.vdb ×	- 0
Index	COLUMNS.INDEX	Book	ooks Project/BooksWebService_VDB.vdb	•
Conter	nt			
G BOC G BOC	NESWEBSERVICE.BOOK NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT NESMIL.BOOKCOLLECT	5. GETBOOKCOLLECTION. GET 10NDCCUMENT. BOOKCOLLEC 10NDCCUMENT. MAPPINGCLA 10NDCCUMENT. MAPPINGCLA 10NDCCUMENT. MAPPINGCLA 10NDCCUMENT. MAPPINGCLA 10NDCCUMENT. MAPPINGCLA 10NDCCUMENT. MAPPINGCLA	GETBOOKCOLLECTIONOUTPUT.MESSAGE SAMPLES mmuuid:c49b5ec1-f5e5-1f59-a112-e0eac778cc3c BooksWeb LECTION. BOOK, BUSN mmuuid: 1646cec4-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument.b LECTION. BOOK, AUTHORS Mmuuid: 1561ed04-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument LECTION. BOOK, AUTHORS mmuuid: 1561ed04-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument LECTION. BOOK, AUTHORS mmuuid: 1561ed04-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument LECTION. BOOK, AUTHORS mmuuid: 1561ed03-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument LECTION. BOOK, PUBLISHINGINFORMATION.PUBLISHDATE mmuuid: 1646cec2-0dca-1eec-8518-c32201e76066 Book LECTION. BOOK, PUBLISHINGINFORMATION.PUBLISHER mmuuid: 1646cec2-0dca-1eec-8518-c32201e76066 Book LECTION. BOOK, SUBLISHINGINFORMATION.PUBLISHER mmuuid: 1646cec2-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument LECTION. BOOK, SUBLISHINGINFORMATION.PUBLISHER mmuuid: 1646cec2-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument. LECTION. BOOK, SUBLISHINGINFORMATION and the tee-8518-c32201e76066 BooksXML.bookCollectionDocument. LECTION. BOOK, SUBLISHINGINFORMATION and the tee-8518-c32201e76066 BooksXML.bookCollectionDocument.bc LECTION. BOOK, SUBLISHINGINFORMATION and the tee-8518-c32201e76066 BooksXML.bookCollectionDocument.bc CLASSES.AUTHOR.AUTHOR mmuuid: 18216c7-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument.bc CLASSES.BOOK.JEBLISHER mmuuid: 18216c7-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument.Map CLASSES.BOOK.PUBLISHER mmuuid: 18216c7-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument.Map CLASSES.BOOK.PUBLISHER mmuuid: 18216c7-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument.Map CLASSES.BOOK.SUBLISHER mmuuid: 18216c7-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument.Map CLASSES.BOOK.PUBLISHER mmuuid: 18216c7-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectionDocument.Map CLASSES.BOOK.SUBLISHER mmuuid: 18216c7-0dca-1eec-8518-c32201e76066 BooksXML.bookCollectio	
< BOC	KSXMLIBOOKLISTINGL	UCCOMENT.BOOKLISTING.BO	BUUK.@ISBN mmuuid:U/bUIC42-Udca-Ieec-8518-C322U1e/6066 BooksXML.bookListingDocument.bookListing.bo	× ×
Overvie	w Problems Indexes	Web Services Execute		

# Figure 5.23. VDB Editor Indexes Panel

# 5.2.4. User Files

You can add additional artifacts or files to your VDB via the User Files tab in the **VDB Editor**. Select the **Add...** button and locate/select files from your local file system.

🔋 PartsVDB.vdb 🗙	- 8
User Files:	
File Name	
PartsResourceDocument.txt	
test_sql.txt	
	odd
	Remove
Overview Problems Indexes User Files Web Services Execute	

Figure 5.24. VDB Editor User Files Panel

# 5.2.5. Web Services

The **Web Services** tab provides a means to specify the target namespace URL for your **WSDL** and to display a preview of that **WSDL**.

🔋 CustomerAccounts.vdb 🛛 🚺 BooksWebService_VDB.vdb 🗙 🔤 🗖
This VDB contains one or more Web Service mo(Books Project/BooksWebService_VDB.vdb) be customized to specify the following information:
Target namespace URI for the WSDL:
http://com.metamatrix/BooksWebService_VDB
Default WSDL namespace: http://schemas.xmlsoap.org/wsdl/ Press the "Preview WSDL" button to display the WSDL file that will be generated for the VDB: Preview WSDL
Overview Problems Indexes Web Services Execute

# Figure 5.25. VDB Editor Web Services Panel



#### Figure 5.26. WSDL Preview Dialog

# 5.2.6. Execute

The Execute tab provides the capability to create and edit connectors for each source model in your VDB and to launch the SQL Explorer to query your data.

For more information on Connectors, see <u>Manage Connectors</u>.

Products_SQL_Server.xmi	Roduct_Views.xmi	- 6
xecution Status		
xecute The virtual databas	se is ready to execute.	
xecution Options		
ansaction AutoWrap: OPTIM	ISTIC 💌	
onnectors		
elect row below to view or edit :	specific connector properties	
Source Model	Connector	Edit
Products_SQL_Server	Products_SQL_Server Connector - SQL Server Connector	
		Unbina
<u> </u>		
Connector Properties:	Show Advanced	Properties
Property	Value	
JDBC URL	jdbc:sqlserver://slntdb25.dat.mycompany.com:1433;databasename=produ	
User Name	products	
Password	**	
Class Path	extensionjar:sqljdbc4.jar	
Connection Source Class	com.microsoft.sqlserver.jdbc.SQLServerDriver	
Is XA	false	

Figure 5.27. VDB Editor Execute Panel

# Importers

#### Go To: Table of Contents [index.html]

The **Import Wizard** provides a means to create a model based on the structure of a data source, to convert existing metadata (i.e. **WSDL** or **XML Schema**) into a source model or to load existing metadata files into the current VDB.

To launch the **Import Wizard**, choose the **File > Import** action or select a project, folder or model in the tree and right-click choose "**Import...**"

E Import	X
Select	
Import metadata from a specially-formatted text file into a new or existing model.	L
Select an import source:	
type filter text	
<ul> <li>▷ ➢ General</li> <li>▷ ➢ CVS</li> <li>▽ ➢ Metadata Modeling</li> </ul>	
😪 Metadata from IBM Rational Rose (c) Model	
■ Metadata from Text Files on file system	
(X) Salesforce as Relational Source Model	Ξ
⊌ Web Service as Relational Source Model	
XML Schema as Relational Source Model	
S XSD Schemas	
👂 🗁 Plug-in Development	_
👂 🗁 Run/Debug	
D 🔁 Team	~
⑦ < Back Next > Einish Cancel	

## Figure 6.1. Import Wizard

# **6.1. Import Archive File**

• You can import files from within an archive file using the steps below.

• Step 1 - In Model Explorer choose the File > Import action

in the toolbar or select a project, folder or model in the tree and choose Import...

- Step 2 Select the import option General > Archive file and click NEXT>
- Step 3 Click on the Browse button to find the archive (i.e. zip) file you wish to import. and click NEXT>

🕾 Import	
Archive file Import the contents of an archive file in zip or tar format from the local file system.	
From archive file: D:\Designer\schema data\ePortfolio.zip	Browse
Image: Construction of the sector of the	
Filter Types Select All Deselect All	
Into folder: WestCoastSales	Browse
Overwrite existing resources without warning	
<pre> &lt; Back Next &gt; Finish</pre>	Cancel

#### Figure 6.2. Import Archive File Dialog

- **Step 4** Open directory structure tree in left panel. Define which files from archive to import by checking/unchecking in trees.
- Step 5 Specify valid Into Folder by editing the text field, or select Browse... button to locate and select a folder or project in your workspace.
- Step 6 Click Finish.

# 6.2. Import From File System

- You can import files located on your local file system using the steps below.
  - Step 1 In Model Explorer choose the File > Import action

in the toolbar or select a project, folder or model in the tree and choose Import...

- Step 2 Select the import option General > File system and click Next>
- Step 3 Click on the Browse button to find the archive (i.e. zip) file you wish to import. and click NEXT>
- **Step 4** Open directory structure tree in left panel. Define which files to import by checking/ unchecking in trees.

🙅 Import	
File system Import resources from the local file system.	
From directory:       D:\Testing\Schemas\Dublin         Image: Comparison of the second se	Browse
Filter Types       Select All       Deselect All         Into folder:       AllCustomerAccounts         Options       Overwrite existing resources without warning         Create complete folder structure       Image: Create selected folders only	Browse
< Back Next > Finish	Cancel

#### Figure 6.3. Import From File System Dialog

- Step 5 Specify valid Into Folder by editing the text field, or select Browse... button to locate and select a folder or project in your workspace.
- Step 6 Modify any Options to customize the imported files.
- Step 7 Click Finish.

# 6.3. Import From JDBC

 You can create relational source models from your JDBC source schema data using the steps below.



#### è

in the toolbar or select a project, folder or model in the tree and choose Import...

 Step 2 - Select the import option Metadata Modeling > Metadata from JDBC Database and click Next>



Figure 6.4. Import Metadata from JDBC Database

• Step 3 - In the Import Database via JDBC dialog, select existing or previous source configuration or press Sources... to create new source configurations.

🐏 JDBC Sour	ce Configurations	
Update JDBC Press the "OK"	C source configurations button to accept your changes.	
-Source Co	nfigurations	
Northwind_C	Dracle	Add
		Remove
		<u>S</u> ave
		Load
Name:	Northwind_Oracle	
Driver:	MetaMatrix Oracle	/ers
URL Syntax:	jdbc:mmx:oracle:// <host>:1521;Sid=<sid></sid></host>	perties
URL:	jdbc:mmx:oracle://slntds01:1521;Sid=ds01a	•
User Name:	northwind	
	Test Connection	
	ОК	Cancel

#### Figure 6.5. Import Source Configurations Dialog

• Step 4 - After selecting Source, input password and press Connect to Database. Press Next> (or Finish> if enabled)

🗠 Import D	atabase via JDBC		
Select a JD Press the "No	BC source configuration ext >" button to continue or the "Finish" button to finish.		
Source Nor Driver: URL: User Name:	thwind_Oracle MetaMatrix Oracle jdbc:mmx:oracle://slntds01:1521;Sid=ds01a	•	Sources
Password:	*******		
	< Back Next > Finish		Cancel

#### Figure 6.6. Import Database via JDBC Dialog

• Step 5 - On the Select Database Metadata page, select the types of objects in the database to import. Press Next> (or Finish> if enabled).

Import Database via JDB Select Database Metadat Press the "Next >" button to con	C a tinue or the "Finish" button to finish.	
<ul> <li>Foreign Keys</li> <li>Indexes</li> <li>Unique Only</li> <li>Approximations Allowed</li> <li>Procedures</li> </ul>	Table Types SYNONYM SYSTEM TABLE TABLE VIEW	Select All Deselect All
	<pre> &lt; Back Next &gt; Finish</pre>	Cancel

#### Figure 6.7. Select Database Metadata Dialog

• Step 6 - On the Select Database Objects page, view the contents of the schema, or change selections. Select which database schema objects will be used to construct relational objects. Press Next> (or Finish> if enabled)

🙅 Import Database via JDBC							
Select Database Objects							
Press the "Next >" button to continue or the "Finish"	button to finish	h.					
Northwind_Oracle	Columns Inde	exes   Primary Key	Imported Fore	eign Keys   Exporte	ed Foreign Keys	Description	
🖻 🗹 TABLE 🛛 🔼	TABLE_CAT	TABLE_SCHEM	TABLE_NAME	COLUMN_NAME	DATA_TYPE	TYPE_NAME	COLUMN
CATEGORIES		NORTHWIND	CUSTOMERS	CUSTOMERID	1	CHAR	
CUSTOMERCUSTOMERDEMO		NORTHWIND	CUSTOMERS	COMPANYNAME	12	VARCHAR2	
CUSTOMERDEMOGRAPHICS		NORTHWIND	CUSTOMERS	CONTACTNAME	12	VARCHAR2	
		NORTHWIND	CUSTOMERS	CONTACTTITLE	12	VARCHAR2	
EMPLOYEES		NORTHWIND	CUSTOMERS	ADDRESS	12	VARCHAR2	
		NORTHWIND	CUSTOMERS	DECTON	12	VARCHARZ	
		NORTHWIND	CUSTOMERS	REGION	12	VARCHARZ	
		NORTHWIND	CUSTOMERS	COUNTRY	12	VARCHAR2	
		NORTHWIND	CUSTOMERS	PHONE	12	VARCHAR2	
		NORTHWIND	CUSTOMERS	FAX	12	VARCHAR2	
			COSTOTIENS		12	THICK HILL	
	2						
< >							
13 table object(c) celected to import							
To caple object(s) selected to import.							
Deselect All							
🔲 Show Only Selected Schemas							
· · · · · · · · · · · · · · · · · · ·							
< Daux Next > Finish Cancel							

#### Figure 6.8. Select Database Options Dialog

• Step 7 - On the Specify Import Options page, specify desired Model Name as well as any other options used to customize the constructed relational objects. Press Finish> to complete.

🕾 Import Database via JDBC	
Specify Import Options	9
Specify the name, location, and import options for the new relational model of database "Northwind_Oracle.xmi". $\square$	
Model Name: Northwind_Oracle.xmi Into Folder: AllCustomerAccounts Update (if existing model selected)	
Include In Model         Select whether the following types of objects should be included in the model:         Image: User	
Model Object Names Select the case-conversion that should be performed when assigning source object names to model objects: None Uppercase C Lowercase	
Source Object Names Select the value of the "Name in Source" property for imported model objects: © Empty © Unqualified Source Names © Fully-Qualified Source Names	
< Back Next > Finish Ca	ancel

#### Figure 6.9. Specify Import Options Dialog

# 6.4. Import Metadata From Text File

- The Teiid Designer provides various import options for translating specific text file metadata into models. This is accomplished via the **Import > Metadata from Text Files on file system** option.
  - Step 1 In Model Explorer choose the File > Import action

in the toolbar or select a project, folder or model in the tree and choose Import...

 Step 2 - Select the import option Metadata Modeling > Metadata from Text Files on file system and click Next>

Import	X			
Select	~			
Import metadata from a specially-formatted text file into a new or existing model.	Ľ			
<u>S</u> elect an import source:				
type filter text				
<ul> <li>▷ General</li> <li>▷ ▷ CVS</li> </ul>	Â			
✓ → Metadata Modeling				
鑬 Metadata from IBM Rational Rose (c) Model				
🗗 Metadata from Text Files on file system				
X Salesforce as Relational Source Model	=			
Web Seprice as Polational Source Model				
WSDL File into Web Service Model				
M XML Schema as Relational Source Model				
S XSD Schemas				
Plug-in Development				
👂 🗁 Run/Debug				
D 🔁 Team	~			
⑦ < <u>Back</u> <u>Next</u> > <u>Finish</u> C	ancel			

#### Figure 6.10. Import Metadata From Text Files

• Step 3 - Select an import type via the drop-down menu shown below.

💀 Import Metadata From Text File	
Select Import Type Select the type of text file import desired.	
Import Type: Relational Tables Relational Tables Purpose: Imp Relationships 1) Sche Relational Virtual Tables Datatypes >> Locations are of the form: ProjectName/FolderName/ModelName/SchemaName.	<b>_</b>

# Figure 6.11. Import Wizard

• These steps required for each type are defined below:

- <u>Relational Tables</u>
- Relational Virtual Tables
- Relationships
- <u>Datatypes</u>

# 6.4.1. Import Relational Tables Text Importer

- To create relational tables from imported text file metadata:
  - Perform Steps 1 thru 3 (above) and select the Relational Tables import type, then click Next >



Figure 6.12. Select Import Type - Relational Tables

• **Step 4** - In the next page, you'll need to provide a source text file containing the metadata formatted to the specifications on the previous page.

🥶 Import Metadata From Text File				×
Select Source Text File and Target Relational Select CSV File to Import and Destination Model or model fo	Model. Ider			
Select Source Text File			<u> </u>	Browse
Target Location Northwind_Oracle_2.xmi				Browse
File Contents				
Cohine				
Use default datatype: string(255)				
F				
	< Back	Next >	Finish	Cancel

#### Figure 6.13. Select Source Text File and Target Relational Model

- **Step 5** Select an existing relational model as the target location for your new relational components using the **Browse...** button to open the Relational Model Selector Dialog. Select a relational model from your workspace or specify a unique name to create a new model.
- Step 6 Select any additional options and choose Finish.

# 6.4.2. Import Relational Virtual Tables Text Importer

- To create relational virtual tables from imported text file metadata:
  - Perform Steps 1 thru 3 (above) and select the Relational Virtual Tables import type, then click Next >

🐵 Import Metadata From Text File	X
Select Import Type	A D
Select the type of text file import desired.	
Import Type: Relational Virtual Tables Purpose: Imports virtual relational tables and procedures from a CSV text file Table data is expected to be of the form: >> TableName, "SQL Statement", Description	e into a relational virtual model.
A sample of typical input data is shown below: Sample File Format:	
VTable1, "SELECT * FROM PMod1.PTab1, PMod1.PTable2", "Descrip, no emb VTable2, "SELECT * FROM PMod1.PTab2", "Descrip 2 with quotes ""xxx""" VTable3, "SELECT * FROM PMod1.PTab3", "Descrip 3" VTable4, "SELECT * FROM PMod1.PTab4", "Descrip 4" VTable5, "SELECT * FROM PMod1.PTab5", "Descrip 5" VTable6, "SELECT * FROM PMod1.PTab5", "Descrip 5" VTable6, "SELECT * FROM PMod1.PTab5", "Descrip 7" VTable7, "SELECT * FROM PMod1.PTab5", "Descrip 7" VTable8, "SELECT * FROM PMod1.PTab6", "Descrip 8" VTable9, "SELECT * FROM PMod1.PTab6", "Descrip 9" VTable9, "SELECT * FROM PMod1.PTab5", "Descrip 10" VTable10, "SELECT * FROM PMod1.PTab10", "Descrip 10" VProc1, "CREATE VIRTUAL PROCEDURE BEGIN SELECT * FROM PMod1.PTab	bedded quotes" b10; END", "Descrip VProc1"
< Back	Next > Finish Cancel

#### Figure 6.14. Select Import Type - Relational Virtual Tables

• **Step 4** - In the next page, you'll need to provide a source text file containing the metadata formatted to the specifications on the previous page.

🙅 Import Metadata From Text File				
Select Source Text File and Target Virtual Select CSV File to Import and Destination Model or mod	Relational Model. lel folder			
Select Source Text File				Browse
Target Location File Contents				Browse
	< Back	Next >	Finish	Cancel

#### Figure 6.15. Select Source Text File and Target Virtual Relational Model

- Step 5 Select an existing relational virtual model as the target location for your new model components using the **Browse...** button to open the Virtual Model Selector Dialog. Select a virtual relational model from your workspace or specify a unique name to create a new model.
- Step 6 Select Finish.

# 6.4.3. Import Relationships Text Importer

- To create relationships from imported text file metadata:
  - Perform Steps 1 thru 3 (above) and select the Relationships import type, then click Next >
| 🐵 Import Metadata From Text File   |             |
|--|-------------|
| Select Import Type   | 2h          |
| Select the type of text file import desired.   | 94          |
| Import Type: Relationships Purpose: Imports relationships from a CSV text file into a relationship model.  | •           |
| Relationship data is expected to be of the form:<br>>> RelationshipName, RelationshipType, SourceRoleObj, TargetRoleObj, "Description"<br>RoleObjs can be path or mmuuid form.   |             |
| A sample of typical input data is shown below:<br>Sample File Format:  |             |
| Rel_1, Generic, Proj1/PModels/P1/T1, Proj1/PModels/P1/T2, "First Description"<br>Rel_2, Generic, Proj1/PModels/P1/T2, Proj1/PModels/P1/T3, "Second Description"<br>Rel_3, Generic, Proj1/PModels/P1/T3, Proj1/PModels/P1/T4, "Third Description"<br>Rel_4, Generalization, Proj1/PModels/P1/T4, Proj1/PModels/P1/T5, "Fourth Description"<br>Rel_5, Proj1/Model1/Types/MyType, Proj1/PModels/P1/T5, Proj1/PModels/P1/T5, "Fifth Description"<br>Rel_6, Generalization, mmuuid:41be0640-c297-1fa5-9c18-fc4a051b425a, Proj1/PModels/P1/T5, "Seventh Description"<br>Rel_7, Generalization, Proj1/PModels/P1/T5, mmuuid:41be0640-c297-1fa5-9c18-fc4a051b425a, "Seventh Description" | 2<br>2<br>2 |
|  |             |
| < Back Next > Finish Car   |             |

### Figure 6.16. Select Import Type - Relationships

• **Step 4** - In the next page, you'll need to provide a source text file containing the metadata formatted to the specifications on the previous page.

🥶 Import Metadata From Text File		
Select Source Text File and Target Relationship Model. Select CSV File to Import and Destination Model or Model Folder		
Select Source Text File		Browse
Target Location		Browse
File Contents		
< Back	Next > Finish	Cancel

### Figure 6.17. Select Source Text File and Target Relationships Model

- **Step 5** Select an existing relationship model as the target location for your new model components using the **Browse...** button to open the Relationship Model Selector Dialog. Select a relationship model from your workspace or specify a unique name to create a new model.
- Step 6 Select Finish.

### 6.4.4. Import Datatypes Text Importer

- To create datatypes from imported text file metadata:
  - Perform Steps 1 thru 3 (above) and select the Datatypes import type, then click Next >

💀 Import Metadata From Text File
Select Import Type
Select the type of text file import desired.
Import Type: Datatypes  Purpose: Imports datatypes from a CSV text file into a datatype model. Datatype data is expected to be of the form:
<ul> <li>&gt; TYPE or ENUM, Name, "Description", baseTypeName, Length, MinLength, MaxLength, MinBound, MinInclusive, MaxBound, MaxInclusive, TotalDigits, FractionDigits. Blank or zero values are ignored.</li> <li>A sample of typical input data is shown below:</li> </ul>
Sample File Format:
TYPETAG,Name,Description,BaseType,Length,MinLength,MaxLength,MinBound,MinInclusive,MaxBound,MaxInclusive,TotalDig TYPE, Atomic1, "Description for Type1", BType, 5, 1, 10, 1, 2, 1, 2, 5, 2 TYPE, Atomic2, "Description for Type2", BType, 5, 1, 10, 1, 2, 1, 2, 5, 2 TYPE, Atomic3, "Description for Type3", BType, 5, 1, 10, 1, 2, 1, 2, 5, 2 TYPE, Atomic4, "Description for Type4", BType, 5, 1, 10, 1, 2, 1, 2, 5, 2 TYPE, Atomic4, "Description for Type4", BType, 5, 1, 10, 1, 2, 1, 2, 5, 2 TYPE, Atomic5, "Description for Type5", BType, 5, 1, 10, 1, 2, 1, 2, 5, 2
< Back Next > Finish Cancel

### Figure 6.18. Select Import Type - Datatypes

• **Step 4** - In the next page, you'll need to provide a source text file containing the metadata formatted to the specifications on the previous page.

🙅 Import Metadata From Text File				
Select Source Text File and Target Datatypes I Select CSV File to Import and Destination Model or model fold	Model. <sup>Jer</sup>			
Select Source Text File			<u> </u>	Browse
Target Location				Browse
File Contents				
	< Back	Next >	Finish	Cancel

### Figure 6.19. Select Source Text File and Datatypes Model

- **Step 5** Select an existing datatype model as the target location for your new model components using the **Browse...** button to open the Datatypes Model Selector Dialog. Select a datatypes model from your workspace or specify a unique name to create a new model.
- Step 6 Select any additional options and choose Finish.

# 6.5. Import WSDL as Relational Source

In addition to creating relational source models by importing your metadata (*Import Relational Data*) you can also turn your **WSDL** into relational models.

- To create relational models from **WSDL** use the steps below.
  - Step 1 In Model Explorer choose the File > Import action

in the toolbar or select a project, folder or model in the tree and choose Import...

 Step 2 - Select the import option Metadata Modeling > Web Service as Relational Source Model and click Next>

E Import	X
Select	
Create Relational Model from Web Service.	2
<u>S</u> elect an import source:	
type filter text	
D General	<u> </u>
p 🔁 cvs	
Metadata Modeling	
Metadata from IBM Rational Rose (c) Model	
Metadata from JDDC Database	
Salesforce as Pelational Source Model	
	=
Web Service as Relational Source Model	
WSDL File into Web Service Model	11
M XML Schema as Relational Source Model	
SI XSD Schemas	
Plug-in Development	
Run/Debug	
D 🔁 Team	~
(?) < Back Next > Finish Cancel	

# Figure 6.20. Import WSDL as Relational Source Model

• Step 3 - On the next page provide a source WSDL file, either in your workspace, local on your file system or provide a valid URL.

Select WSDL So	urce	
Workspace	file:/D:/Designer/workspace/WSDL_Example/mywsdls/TerraService.asmx.wsdl	Browse
C File System		Browse
C URL		
	V	alidate WSDL
Select Target R	elational Model	
Model Name:	TerraServiceSource	Browse
		e 123

### Figure 6.21. Source and Target Selection Dialog

- Step 4 When WSDL is defined, you'll be required to validate your WSDL via the Validate WSDL button (see above figure).
- Step 5 Specify a unique target relational model name in the Model Name field, or select an existing relational model in your workspace via the Browse... button. Click Next> when model name is defined.
- Step 6 Specify a unique target location in the Location field (i.e. project or folder) for the relational model or select an existing location in your workspace via the Browse... button. Click Next> when location is defined.
- Step 7 The next page allows you to selected individual Web Service Operations from which to create relational tables. The default behavior of this page selects all available operations in the tree. For WSDLs backed with large schemas, there may be numerous operations which may result in a very large number of tables being generated. In this case, you may wish choose a limited number of operations at a time and repeat this process using the same target relational model.
  - To select all of the XSD files in the directory, click the checkbox next to the folder in the left panel.

 To select individual XSD files, click the checkboxes next to the files you want in the right panel



### Figure 6.22. WSDL Operations Selection Dialog

• Step 5 - Click Finish. After generation the new model can be found in the specified location in your workspace..

# 6.6. Import WSDL Into Web Service

You can create a **Web Service** model by selecting a **WSDL** file in your workspace, importing **WSDL** files from the file system or by defining a URL. The Teiid Designer will interpret the **WSDL**,

locate any associated or dependent **XML Schema** files, generate an **XML View** of the schema components and create a **Web Service** model representing the interfaces and operations defined in the **WSDL**.

- There are three options for selecting the WSDL for your Web Service generation
  - Workspace Location
  - File System Location
  - URL

Detailed steps for each of these options is described below, as well as a description of how the wizard handles **WSDL** errors.

## 6.6.1. Import WSDL From Workspace Location

- You can create a Web Service model by selecting a WSDL file from your workspace.
  - Step 1 In Model Explorer choose the File > Import action

in the toolbar or select a project, folder or model in the tree and choose Import...

 Step 2 - Select the import option Metadata Modeling > WSDL File into Web Service Model and click Next>

E Import	X
Select	
Create Web Service Model from WSDL file from file system.	5
<u>S</u> elect an import source:	
type filter text	
Dependence General	_
CVS	
🗢 🗁 Metadata Modeling	
😪 Metadata from IBM Rational Rose (c) Model	
🗍 Metadata from JDBC Database	
Metadata from Text Files on file system	
X Salesforce as Relational Source Model	Ξ
🗹 UDF Model	
By Web Service as Relational Source Model	
💕 WSDL File into Web Service Model	
👸 XML Schema as Relational Source Model	
S XSD Schemas	
👂 🗁 Plug-in Development	
👂 🧁 Run/Debug	
D 🔁 Team	~
(?) < Back Next > Finish Cancel	

# Figure 6.23. WSDL File Into Web Service Dialog

• Step 3 - Input a valid name for your Web Service model and select the Workspace... button. Locate your workspace WSDL file in the selection dialog and click OK>.

WSDL File Selection Press the "Next >" button to continue.		-
	0	)
Web Service Model Name CustomerWS		
Workspace File System URL	121211	
WSDL Files	×	
D:\MetaMatrix\EnterpriseKits\5.5_Release\R050500-081407\workspace\WSDL_Example\mywsdls\TerraService.asmx.wsdl		
< Back Next > Finish C	ancel	1

Figure 6.24. WSDL File Selection Dialog

🙅 WSDL File Selection	
Choose a WSDL file to add: AllCustomerAccounts EastCoastSales WSDL_Example mywsdls TerraService.asmx.wsdl WestCoastSales	
ОК	Cancel

### Figure 6.25. WSDL File Workspace Selection Dialog

i	Note
	<ul> <li>If no WSDL is selected or specified then the importer will only create an empty Web Service model. No XML Schema or XML View models will be generated.</li> </ul>
	<ul> <li>Any referenced files (WSDLs or schemas) must either be embedded in the WSDL file or exist on your file system.</li> </ul>

• Step 4 - The next page is titled Namespace Resolution. This page identifies successful and errant WSDL namespace resolution. The main WSDL document will essentially always be resolved, since the workspace file chooser is used to obtain the path. Problems will occur when the main WSDL file imports other WSDL files that cannot be resolved. If no errors, select Next to proceed, or Finish (if enabled) to complete with default options.

iesp	paces	E ·
2	Namespace	Path

### Figure 6.26. Namespace Resolution Dialog

• Step 5 - The next page WSDL Operations Selection allows customizing the resulting content of your Web Service model by selecting/deselecting various operations and interfaces in the following dialog.



### Figure 6.27. Namespace Resolution Dialog

• Step 6 - The next page is titled Schema Workspace Location Selection. This page lists all schemas imported by the WSDL (along with any dependent schemas referenced within schemas) as well as schemas embedded in the WSDL and indicates whether or not they are resolvable. All resolved schemas will be created in a separate file and added to the workspace. The editor panel allows you to change the default file name of the new schema file(s).

If no errors, select Next to proceed, or Finish to complete with default option

Creat	te Web Service from WSDL File		
n <b>em</b> a I worke	a Workspace Location Selection space schema locations are valid. Click "Next	" to continue.	
chema	a Location Information		-
-	Namespace	Source File	Target Path
	http://terraserver-usa.com/terraserver/	D:\Designer\workspace\WSDL_E	AllCustomerAccounts\TerraService.asmx.xsd
j Edil Fol	itor Ider:		)
			Apply

### Figure 6.28. Namespace Resolution Dialog

• Step 7 - The last page titled XML Model Generation allows you to change the name of the XML View model if the Generate virtual XML document model is checked. Input desired name or use the default name provide. Select Finish to complete.

🗠 Create V	Veb Service from WSDL File	
XML Model Ready to ge	I Generation	•
Generate Location: XML Model:	e virtual XML document model VWSDL_Example CustomerWSResponses	]
	<pre>&lt; Back Next &gt; Finish Can</pre>	cel

### Figure 6.29. Namespace Resolution Dialog

## 6.6.2. Import WSDL From File System Location

- You can create a Web Service model by selecting a WSDL file from your local file system.
  - Step 1 In Model Explorer choose the File > Import action

in the toolbar or select a project, folder or model in the tree and choose Import...

 Step 2 - Select the import option Metadata Modeling > WSDL File into Web Service Model and click Next>

E Import	X
Select	
Create Web Service Model from WSDL file from file system.	5
<u>S</u> elect an import source:	
type filter text	
Dependence General	_
CVS	
🗢 🗁 Metadata Modeling	
😪 Metadata from IBM Rational Rose (c) Model	
🗍 Metadata from JDBC Database	
Metadata from Text Files on file system	
X Salesforce as Relational Source Model	Ξ
🗹 UDF Model	
By Web Service as Relational Source Model	
💕 WSDL File into Web Service Model	
👸 XML Schema as Relational Source Model	
S XSD Schemas	
👂 🗁 Plug-in Development	
👂 🧁 Run/Debug	
D 🔁 Team	~
(?) < Back Next > Finish Cancel	

# Figure 6.30. WSDL File Into Web Service Dialog

• Step 3 - Input a valid name for your Web Service model and select the File System... button. Locate your file system WSDL file in the selection dialog and click OK>.

🕾 Create Web Service from WSDL File
WSDL File Selection Select "Next" to resolve namespaces referenced by the 1 selected WSDL(s).
Web Service Model Name CustomerWS Workspace File System URL
WSDL Files
< Back Next > Finish Cancel

### Figure 6.31. WSDL File Selection Dialog

# Note • If no WSDL is selected or specified then the importer will only create an empty Web Service model. No XML Schema or XML View models will be generated. • Any referenced files (WSDLs or schemas) must either be embedded in the WSDL file or exist on your file system.

• Step 4 - The next page is titled Namespace Resolution. This page identifies successful and errant WSDL namespace resolution. The main WSDL document will essentially always be resolved, since the workspace file chooser is used to obtain the path. Problems will occur when the main WSDL file imports other WSDL files that cannot be resolved. If no errors, select Next to proceed, or Finish (if enabled) to complete with default options.

mesp	aces	
-	Namespace	Path

### Figure 6.32. Namespace Resolution Dialog

• Step 5 - The next page WSDL Operations Selection allows customizing the resulting content of your Web Service model by selecting/deselecting various operations and interfaces in the following dialog.



### Figure 6.33. Namespace Resolution Dialog

• Step 6 - The next page is titled Schema Workspace Location Selection. This page lists all schemas imported by the WSDL (along with any dependent schemas referenced within schemas) as well as schemas embedded in the WSDL and indicates whether or not they are resolvable. All resolved schemas will be created in a separate file and added to the workspace. The editor panel allows you to change the default file name of the new schema file(s).

If no errors, select Next to proceed, or Finish to complete with default option

Schema Workspace Location Selection   All workspace schema locations are valid. Click "Next" to continue.     Schema Location Information     Namespace   Namespace   Source File   http://terraserver-usa.com/terraser   C:\DOCUME~     Editor   Folder:   Name (without file extension):	
All workspace schema locations are valid. Click "Next" to continue.  Schema Location Information  Namespace Source File http://terraserver-usa.com/terraser C:\DOCUME~ Editor Folder: Name (without file extension):	
Schema Location Information         Namespace       Source File         http://terraserver-usa.com/terraser       C:\DOCUME~         Editor       Folder:         Name (without file extension):       Name (without file extension):	
Schema Location Information          Namespace       Source File         http://terraserver-usa.com/terraser       C:\DOCUME~         Editor       Folder:         Name (without file extension):       Name (without file extension):	
Namespace       Source File         http://terraserver-usa.com/terraser       C:\DOCUME~         Editor       Image: Comparison of the second sec	ľ
http://terraserver-usa.com/terraser       C:\DOCUME~         Editor	Target Path
Editor         Folder:         Name (without file extension):	MyProj\_TerraService.asmx4112.xsd
	Apply

### Figure 6.34. Namespace Resolution Dialog

• Step 7 - The last page titled XML Model Generation allows you to change the name of the XML View model if the Generate virtual XML document model is checked. Input desired name or use the default name provide. Select Finish to complete.

🗠 Create V	Veb Service from WSDL File	
XML Model Ready to ge	I Generation	•
Generate Location: XML Model:	e virtual XML document model VWSDL_Example CustomerWSResponses	]
	<pre>&lt; Back Next &gt; Finish Can</pre>	cel

### Figure 6.35. Namespace Resolution Dialog

# 6.6.3. Import WSDL From URL

- You can create a **Web Service** model by selecting a **WSDL** file based on a URL.
  - Step 1 In Model Explorer choose the File > Import action

in the toolbar or select a project, folder or model in the tree and choose Import...

 Step 2 - Select the import option Metadata Modeling > WSDL File into Web Service Model and click Next>

E Import	X
Select	
Create Web Service Model from WSDL file from file system.	L
<u>S</u> elect an import source:	
type filter text	
D 🎘 General	^
CVS	
🗢 🗁 Metadata Modeling	
😪 Metadata from IBM Rational Rose (c) Model	
📜 Metadata from JDBC Database	
Metadata from Text Files on file system	
X Salesforce as Relational Source Model	=
🗃 UDF Model	
🗗 Web Service as Relational Source Model	
By WSDL File into Web Service Model	
🕉 XML Schema as Relational Source Model	
S XSD Schemas	
Plug-in Development	-
Run/Debug	
D 🔁 Team	~
2 _ Back Next > Enish Cancel	

# Figure 6.36. WSDL File Into Web Service Dialog

• Step 3 - Input a valid name for your Web Service model and select the URL... button. Locate your file system WSDL file in the selection dialog and click OK>.

🙅 Create Web Service from WSDL File	
<b>WSDL File Selection</b> Press the "Next >" button to continue.	<b>(</b>
Web Service Model Name WS_FromWSDL Workspace File System URL	
WSDL Files http://terraservice.net/TerraService.asmx?WSDL	
< Back Next > Finish	Cancel

### Figure 6.37. WSDL File Selection Dialog

# Note • If no WSDL is selected or specified then the importer will only create an empty Web Service model. No XML Schema or XML View models will be generated. • Any referenced files (WSDLs or schemas) must either be embedded in the WSDL file or exist on your file system.

• Step 4 - The next page is titled Namespace Resolution. This page identifies successful and errant WSDL namespace resolution. The main WSDL document will essentially always be resolved, since the workspace file chooser is used to obtain the path. Problems will occur when the main WSDL file imports other WSDL files that cannot be resolved. If no errors, select Next to proceed, or Finish (if enabled) to complete with default options.

Namespace	Path
http://terraserver-usa.com/terraserver/	http://terraservice.net/TerraService.asmx?WSDL

### Figure 6.38. Namespace Resolution Dialog

• Step 5 - The next page WSDL Operations Selection allows customizing the resulting content of your Web Service model by selecting/deselecting various operations and interfaces in the following dialog.



### Figure 6.39. Namespace Resolution Dialog

• Step 6 - The next page is titled Schema Workspace Location Selection. This page lists all schemas imported by the WSDL (along with any dependent schemas referenced within schemas) as well as schemas embedded in the WSDL and indicates whether or not they are resolvable. All resolved schemas will be created in a separate file and added to the workspace. The editor panel allows you to change the default file name of the new schema file(s).

If no errors, select Next to proceed, or Finish to complete with default option

reate Web Service fro	om WSDL File			
ema Workspace Loca	ation Selection			
vorkspace schema locations	; are valid. Click "Nex	«t" to continue.		
ema Location Information				ľ
🗑 🛛 Namespace		Source File	Target Path	
http://terraserver-u	sa.com/terraser	C:\DOCUME~	MyProj\_TerraServic	e.asmx4112.xsd
Editor				×
Editor Folder:				×
Editor Folder: Name (without file extens	ion):			×
' Editor Folder: Name (without file extens	ion):			×
<sup>'</sup> Editor Folder: Name (without file extens	ion):			×  Apply
<sup>'</sup> Editor Folder: Name (without file extens	ion):			× 
' Editor Folder: Name (without file extens	ion):			× 
Èditor Folder: Name (without	file extens	file extension):	file extension):	file extension):

### Figure 6.40. Namespace Resolution Dialog

• Step 7 - The last page titled XML Model Generation allows you to change the name of the XML View model if the Generate virtual XML document model is checked. Input desired name or use the default name provide. Select Finish to complete.

🗠 Create V	Veb Service from WSDL File	×
XML Model Ready to get	I Generation nerate the Web service model	)
Generate Location: XML Model:	virtual XML document model VWSDL_Example CustomerWSResponses	
	< Back Next > Finish Cancel	

### Figure 6.41. Namespace Resolution Dialog

# 6.7. XML Schema as Relational Source

In addition to creating relational source models by importing your metadata you can also turn your existing **XML Schema** into relational source or view models. You can create **XML View** models using the steps below.

- You can import files located on your local file system using the steps below.
  - Step 1 Choose the File > Import action

```
è
```

in the toolbar or select a project, folder or model in the tree and choose **Import...** to launch the **Import Wizard** 

 Step 2 - Select the import option Metadata Modeling > XML Schema as Relational Source and click Next >

E Import X
Select Import an XML Schema file as a Relational Source Model
<u>S</u> elect an import source:
type filter text
<ul> <li>▷ CVS</li> <li>&gt; Metadata Modeling</li> <li>▲ Metadata from IBM Rational Rose (c) Model</li> <li>▲ Metadata from JDBC Database</li> <li>▲ Metadata from Text Files on file system</li> <li>▲ Salesforce as Relational Source Model</li> <li>▲ UDF Model</li> <li>▲ WSDL File into Web Service Model</li> <li>▲ WSDL File into Web Service Model</li> <li>▲ XML Schema as Relational Source Model</li> <li>▲ XML Schema as Relational Source Model</li> <li>▲ XML Schema as Relational Source Model</li> <li>▲ Team</li> </ul>
⑦ < <u>Back</u> <u>Next</u> > <u>Finish</u> Cancel

# Figure 6.42. Select XML Schema As Relational Source Model Option

 Step 3 - On the next page, specify all options for defining document source and catalog options then click Next >

🏧 Import XML Schema as a Relational Source Model 🛛 🛛 🔀
Introduction
Choose the kind of model to create.
Select the type of the XML Document source.
Occument on disk (e.g. file or directory of files)
C Document over HTTP with no request parameters
C Document over HTTP with name-value pair request parameters
C Document over HTTP with an XML request document
Catalog Options
Select the naming scheme for your catalogs
• No catalog
C Use namespaces for catalogs
O Use file names for catalogs
C Use specified name for catalog:
< Back Next > Finish Cancel

### Figure 6.43. WSDL File Into Web Service Introduction Dialog

• Step 4 - On the XML Schema File Selection page, choose a schema file located in your Workspace... or a file on your local File System... . Click Next >

Import XML Schema as a Relational Source Model ML Schema File Selection Select "Next" to resolve namespaces referenced by the 1 selected XSD(s).		
Selected XML Schema Files D:\Testing\Schemas\Employees\Employees.xsd		Workspace File System
< Back Next >	Finish	Cancel

# Figure 6.44. XML Schema File Selection Dialog

 Step 5 - Select global elements that will be used as root elements in your XML documents and move them to the Document Root Elements list. Change any Mapping Options. Click Next >

🙅 Import XML Schema as a Relational Source Model	
Select Root Elements	
Select elements that will be used as root elements in XML docume Selecting elements that are not used will result in unnecessarily o	ints. omplicated models and XPaths.
Select one or more Schema Root Elements to generate Relation	al Models
All Global Elements	Document Root Elements
대 Address 대 Street 대 City 대 State 대 EmpId 대 Employee 대 Name 대 LastName 대 FirstName	Employee SimpleEmployees
	Reset Keep sorted alphabetically
Mapping Options	
• Use "String" mapping type (for best performance querying X	ML in the integration server)
O Use XML types from the Schema (for control of type conversion)	sions within transformations)
	< Back Next > Finish Cancel

# Figure 6.45. XML Schema File Selection Dialog

• Step 6 - Select or deselect schema elements to use in creating relational tables. Click Next >

🐏 Import XML Schema as a Relational Source A	Ao de l	X
<ul> <li>Import XML Schema as a Relational Source of Select Relational Tables</li> <li>Select Relational Tables</li> <li>Select the objects to import from database "Employees".</li> <li>Employees</li> <li>Global</li> <li>ELEMENT</li> <li>Address</li> <li>Employee Employee Type</li> <li>Mame</li> <li>Phone</li> <li>Supervisor_Employee Type</li> <li>SimpleEmployees</li> <li>Employees</li> <li>Supervisor_SimpleEmployee Type</li> <li>Supervisor_SimpleEmployee Type</li> <li>Supervisor_SimpleEmployee Type</li> <li>Supervisor_SimpleEmployee Type</li> <li>Supervisor_SimpleEmployee Type</li> </ul>	Columns Indexes Primary Key Imported Foreign *3 TABLE_CAT TABLE_SCHEM TABLE_NAME COLUMN_NAME D Global Employee mmid	AT
	< Back Next > Finish Cancel	

# Figure 6.46. Select Relational Tables Dialog

• Step 7 - Provide unique model name, model location (project or folder) and options. Click Finish

🐏 Import XI	ML Schema as a Relational Source Model	×
Specify Imp Specify the na	ort Options ame, location, and import options for the new relational model of database "Employees.xmi".	1.5
Model Name: Into Folder: Update (if <b>Model Ob</b> ) Select the ca	Employees.xmi Books Project existing model selected) ject Names ase-conversion that should be performed when assigning source object names to model objects:	:::::::::::::::::::::::::::::::::::::::
C Uppercas	ie ie	
	< Back Next > Finish Cance	*

Figure 6.47. Specify Import Options Dialog

# 6.8. XSD Schema File

- You can import XML Schema file (XSD) files using the steps below.
  - Step 1 In Model Explorer choose the File > Import action

in the toolbar or select a project, folder or model in the tree and choose Import...

 Step 2 - Select the import option Metadata Modeling > XSD Schema files on file system and click Next>

🐏 Import XSD Files	
Import XSD Schemas Select how to import the XSD schemas	SP
C Import XSD schemas from file system C Import XSD schemas via URL	
<pre>&lt; Back Next &gt; Finish</pre>	Cancel

### Figure 6.48. Import XSD Schemas Dialog

- Step 3 Select either Import XSD Schemas from file system or Import XSD Schemas via URL and click Next >
- Step 4a If importing from file system, the Import XSD Files dialog is displayed. Click on the Browse button to find the directory that contains the XSD file(s) you wish to import.
  - To select all of the XSD files in the directory, click the checkbox next to the folder in the left panel.
  - To select individual XSD files, click the checkboxes next to the files you want in the right panel

🙅 Import XSD Files		
File system Import resources from the local file system.		SP
From directory: E:\Testing\Schemas\ePortfolio	¥	Browse
🔁 🗁 ePortfolio	<ul> <li>Gimsassert_v1p0.xsd</li> <li>Gimsparticipation_v1p0.xsd</li> <li>Gimsportfoliocp_v1p0.xsd</li> <li>Gimsreflex_v1p0.xsd</li> <li>Gimsrubric_v1p0.xsd</li> </ul>	
Filter Types     Select All     Deselect All       Into folder:     EmployeesProj		Browse
Options		
Coverwrite existing resources without warning		
C Create complete folder structure		
Greate selected folders only		
< Back	Next > Finish	Cancel

### Figure 6.49. Select XSD From File System

• **Step 4b** - If importing from URL, enter valid XSD Schema URL and click OK to display the final Add XSD Schema URL's wizard page displaying the schemas available for import.

🙅 XSD Schema Url	×
Enter XSD schema URL: http://ns.hr-xml.org/2_4/HR-XML-2_4/StandAlone/Resume.xsd	
Optional User Name Password Verify Hostname (HTTP5)	
OK Cancel	

Figure 6.50. Define XSD Schema URL

🐢 Import XSD Files			
Add XSD Schemas URLs	SP		
XSD schema URLs			
Into folder: EmployeesProj Browse Options Overwrite existing resources without warning Add Dependent Schema Files			
< Back Next > Finish	Cancel		

Figure 6.51. Add XSD Schema URLs
#### • Step 5 - Click Finish.



### Note

XSD files may have dependent files. This importer will determine these dependencies and import these as well if Add Dependent Schema Files is checked

### 6.9. Import Rational Rose Model

The Teiid Designer provides an option to an importer for files exported from Rational Rose versions 2000 and 2001.

The importer creates UML models in the workspace. Supported structural features of UML will be defined by the UMLClasses Metamodel Project. The content of the Rose model file that matches these supported features of the UML metamodel are imported.

The importer allows for subsequent re-importing of the Rose model file which will incorporate changes into the UML model. Any properties of the supported UML constructs that are unique to your use of Rose are implemented using the Teiid Designer's metadata extension functionality.

- To import an Rose file, follow the steps below.
  - Step 1 In Model Explorer choose the File > Import action

in the toolbar or select a project, folder or model in the tree and choose Import...

 Step 2 - Select the import option Metadata Modeling > Metadata from IBM Rational Rose Model and click Next> to open the Rose model selection dialog.

ontained Rose	Units		8
Name	40	File Path	
		D:\Designer\import data\CatML.mdi	
			Check All
			Uncheck All
			Check Children
			Uncheck Children

### Figure 6.52. Rose Model Selection Dialog

• Step 3 - Click on the

button to locate and select your **IBM Rational Rose Model** (\*.mdl or \*.cat) file. Check the appropriate models and/or model children to import. Click **Next >** 

• Step 4 - Select the target models and folders in editor panel and click Next > to display a target models and folders dialog. A progress monitor will display intermediate processing status.

# i Note

If preserving the file system structure of the source Rose Units being imported is desired, a similar file system structure must exist in the model target location prior to starting the wizard.

This wizard page consists of two distinct areas: the selected source Rose Units table and an Editor.

The table identifies the Rose Units being imported. For each Rose Unit a target folder and target model name is shown, as well as, an error or warning indicator. The initial value of the target folder will be set to the model project or folder that was selected prior to starting the wizard (or none if there was no previous selection). Selection buttons to the right of the table aid in selecting table rows. Values in the selected table rows are changed via the editor.

The Editor allows target folders and model names to be changed. The editor modifies all selected table rows (the top area) with its information once the Apply button is selected. If more than one row is selected, the information area shows only the common information.

For example, if two rows are selected that have the same target folder, then that target folder will show in the editor. However, if two rows are selected with different target folders, the editor will not show any information for the folder.

When changes are made in the editor, the appropriate checkbox is automatically checked. Unchecking the checkbox will reset the editor back to the original value. For both the target folder and target model properties, using their respective browse buttons can set new values. This allows navigation of the workspace. Selecting a recently used value in the drop-down dialog can also set new values. The editor can be closed and opened either by using the editor's toolbar button or the top areas toolbar button.

Rose Linit	1	Eolder	UML Model		-
CatML		AllCustomerAccounts	CatML_UML	 	
					Select All
					Unselect All
					Select Children
					Sync Folders
f Editor (1	unit)			 	 
a (-	-	AllCustomerAccounts			
Folder:	A				
Folder:	A Jel:	CatML_UML			<b>•</b> .

### Figure 6.53. Models and Folders Dialog

• **Step 5** - Selecting this import option gives you the opportunity to de-select specific UML model entity(s). All checked elements will be included in your UML model.

-	Rose Importer	
M	Iodel Element Change Verification Uncheck UML model elements that shouldn't be changed. Click "Finish" to execute the changes.	00
1	Pending Changes	
	Old Model ( 0L0):	
	New Model ( NEw):	_
	Only In Old Model ( 0LD); 0 Changes (♠); 0 Only In New Model ( NEW); 3	
	Model Elements	<u>\$</u> {}
	International State Sta	
	Properties	
	importMetadata - this selection is an Addition	
	<back< td="">     Next &gt;     Finish</back<>	Cancel

### Figure 6.54. UML Models Change Dialog

• Step 6 - Click Finish.

## **New Model Wizards**

#### Go To: Table of Contents [index.html]

Models are the primary resource used by the Teiid Designer. Creating models can be accomplished by either directly importing existing metadata or by creating them using one of several **New Model** wizard options. This section describes these wizards in detail.

- The Teiid Designer currently supports the following types of models:
  - - **Relational View**
  - 🤹 XML Service Source

  - ML

     XML

     Document
  - <u>XML Schema (.xsd)</u>

Web Service

Model Extension

**E** <u>Relationship</u>



Use one of the following options to launch the New Model Wizard.

- Choose the File > New... > Metadata Model action
- Select a project or folder in the Section 4.1, "Model Explorer View" and choose the same action in the right-click menu.

<ul> <li>Select</li> </ul>	the	New	button	on	the	main	toolbar
- 23							
and	select		the	Metadata		Model	action
ð							



.

### Note

Model names are required to be unique within Designer. When specifying model names in new model wizards and dialogs error messages will be presented and you will prevented from entering an existing name.

🤷 New Mode	el Wizard	
New Model V Specify model	<b>Wizard</b> name and options then create model file.	P
Location:	AllCustomerAccounts	Browse
Model Name: Model Class:	Relational	
Model Type:	View Model	
Select a mode	el builder (optional):	
중 Generate 중 Transform 중 Copy fro	e from existing UML Models m from an existing model m an existing model of the same model class	
	< Back Next > Finish	Cancel

### Figure 7.1. Import Wizard Selection Dialog

### 7.1. Creating New Relational Source Model

- To create a new empty relational source model:
  - Step 1 Launch the New Model wizard.
  - Step 2 Specify a unique model name.
  - Step 3 Select Relational option from Model Class drop-down menu.
  - Step 4 Select Source Model from Model Type drop-down menu.
  - Step 5 Click Finish.



### Note

You can change the target location (i.e. project or folder) by selecting the **Browse...** button and selecting a project or folder within your workspace.

- In addition to creating a new empty relational source model, the following builder options are available:
  - Copy from existing model of the same model class.
  - Generate from existing UML models.

### 7.1.1. Copy From Existing Model

This builder option performs a structural copy of the contents of an existing model to a newly defined model. You can choose a full copy or select individual model components for copy.

- To create a new relational model by copying contents from another relational source model, complete <u>steps 1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Copy from existing model of the same model class and click Next >. The Copy Existing Model dialog will be displayed.
  - Step 6 Select an existing relational model from the workspace using the browse button.

...

# i

Note

An existing model will be pre-selected if a relational model in the workspace is selected in the Section 4.1, "Model Explorer View" prior to starting the new model wizard.

• Step 7 - Check the Copy all descriptions option if desired. Click Finish

New Model Wizard	×
Copy an Existing Model	
Select an existing model, then select model features to be copied.	
Existing Model: Parts.xmi	
Model Contents	
V B PARTS	
A PART_ID : string(4)	
A PART_NAME : string(255)	
A PART_COLOR : string(30)	
A PART_WEIGHT : string(255)	=
P PK_PARTS	
D SHIP_VIA	
D STATUS	
D SUPPLIER PARTS	
Copy all descriptions	
(?) < Back Next > Finish Ca	ncel

### Figure 7.2. Copy An Existing Model Dialog

### 7.1.2. Generate From Existing UML Model

In the Teiid Designer you can also turn your existing UML models into relational source or view models using the steps below. Note that one option of this builder is to generate relationships between the UML constructs and their generated Relational counterparts.

- To create a new relational model generated from an existing UML model, complete <u>steps 1</u> <u>through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Generate from existing UML models and click Next >.
  - Step 6 On the next page, select the UML model you wish to convert and check/uncheck desired contents, then click Next >.

🗠 New Model Wizard	$\mathbf{X}$
Generate a Model from an Existing UML Model Press the "Next >" button to continue.	-
Select the source UML Model features:         Image: State Stat	
< Back Next > Finish Cancel	

### Figure 7.3. Select UML Features Dialog

Step 7 - On the next page select or create a relationship model to place your generated relationships in. If you do not wish to generate relationships, check the Do not create relationships to UML source entities option. When finished, click Next > or Finish to generate using default options.

🙅 New Model Wizard	
Generate a Model from an Existing UML Model	11
Press the "Next >" button to continue or the "Finish" button to finish.	1
Relationship Options     Create relationships to UML source entities     Select a Relationships Model:     CustomerBolationships write	
	<u></u>
Create a new Relationships model:	Create
<pre> &lt; Back Next &gt;</pre>	Cancel

### Figure 7.4. Relationship Options Dialog

 Step 8 - The next page allows you to choose and customize various optional properties used in generating your relationship model entities. When finished with these options, click Next > or Finish to generate using default options.

🙅 New Model Wizard 🛛 👔
Generate a Model from an Existing UML Model
Press the "Next >" button to continue or the "Finish" button to finish.
- General Ontions
Package Usage: Ignore
Default Relational Column Type:
Length of Type-Defaulted Columns: 10
Class Stereotypes to Ignore
Add
Remove
Class Stereotypes to Set Read Only
Add
Remove
Reachability Constraint: Generate all Reachable
<pre></pre>

### Figure 7.5. UML Generation Options Dialog

• **Step 9** - The next page allows you to choose and customize various optional datatype properties used in generating your relationship model entities. When finished with these options, click **Next >** or **Finish** to generate using default options.

🙅 New Model Wizard	×
Generate a Model from an Existing UML Model Press the "Next >" button to continue or the "Finish" button to finish.	4
Select the Datatype sources          Select the Datatype sources         AllCustomerAccounts         EastCoastSales         System         StocastSales         System         StocastSales         System         StocastSales         System         StocastSales         StocastSales         StocastSales         StocastSales         StocastSales         StocastSales         StocastSales         StocastSales         StocastSales         StocastSales	
< Back Next > Finish Cance	<u>;</u>

### Figure 7.6. Datatype Properties Options Dialog

 Step 10 - The next page allows you to choose and customize various optional properties used in generating your relationship model entities. When finished with these options, click Next > or Finish to generate using default options.

🐢 New Model Wizard		×
Generate a Model from an E Press the "Finish" button to finish.	Existing UML Model	
Generated Key Options – Key Column Base Name: Key Column Type: Key Column Length: Number of Key Columns: Primary Key Stereotype names:	KeyColumn   Iong : xs:integer     I     Primary Key Stereotypes     Add   Remove	
< Ba	ack Next > Finish Cancel	

### Figure 7.7. Generated Key Options Dialog

• Step 11 - Click Finish.

### 7.2. Creating New Relational View Model

- To create a new empty relational view model:
  - Step 1 Launch the New Model wizard.
  - Step 2 Specify a unique model name.
  - Step 3 Select Relational option from Model Class drop-down menu.
  - Step 4 Select View Model from Model Type drop-down menu.

• Step 5 - Click Finish.



### Note

You can change the target location (i.e. project or folder) by selecting the **Browse...** button and selecting a project or folder within your workspace.

- In addition to creating a new empty relational view model, the following builder options are available:
  - Copy from existing model of the same model class.
  - Transform from existing model.
  - Generate from existing UML models.

### 7.2.1. Copy From Existing Model

This builder option performs a structural copy of the contents of an existing model to a newly defined model. You can choose a full copy or select individual model components for copy.

- To create a new relational model by copying contents from another relational view model, complete <u>steps 1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Copy from existing model of the same model class and click Next >. The Copy Existing Model dialog will be displayed.
  - Step 6 Select an existing relational model from the workspace using the browse button.



• Step 7 - Check the Copy all descriptions option if desired. Click Finish

€	New Model Wizard	X
Cop	y an Existing Model	
Sele	ect an existing model, then select model features to be copied.	
Exist	ting Model: Parts.xmi	
Mod	del Contents	_
	B PARTS	^
	ART_ID : string(4)	
	A PART_NAME : string(255)	
	A PART_COLOR : string(30)	
	A PART_WEIGHT : string(255)	=
	P PK_PARTS	
⊳	SHIP_VIA	
⊳	status	
⊳	SUPPLIER	
⊳	SUPPLIER PARTS	~
	Conv all descriptions	_
	copy an acceptions	
		_
?	< <u>B</u> ack <u>N</u> ext > <u>Finish</u> Cancel	

### Figure 7.8. Copy An Existing Model Dialog

### 7.2.2. Transform From Existing Model

This option is only applicable for creating a relational view model from a relational source model with the added feature of creating default transformations (SELECT \* FROM SourceModel.Table\_X) for each source table. The steps are the same as for the <u>Copy Model</u> <u>From Existing Model</u> described above.

There is an additional option in the second page of the wizard which can automatically set the relational table's supports update property to false. If this is unchecked the default value will be true.

### 7.2.3. Generate From Existing UML Model

In the Teiid Designer you can also turn your existing UML models into relational source or view models. Creating a relational view model utilizes the <u>same steps as for relational source models</u> with the one change of setting **Model Type** to **View Model** instead of **Source Model**.

### 7.2.4. Create From XML Schema

In the Teiid Designer you can use your existing schema models to generate relational view tables.

- To create a new relational view model from XML schema:
  - Step 1 Select any XML Schema model (XSD) in the Section 4.1, "Model Explorer View" tree.
  - Step 2 Right-click to display the context menu and select the Modeling > Create Relational View Model from Schema action.
  - Step 3 In the Create Virtual Tables from XSD Schema Wizard dialog, specify a relational view model name.

.ocation: Model Name:	Books Project\ Books			Browse
Global Entiti	es		Entites used to create Tables	
e bookList e bookCol s AudioBc s Bibliogra s Authors s BookNe: s BookSet s BookSet	ing : BooksNS:BookSetFl lection : BooksNS:BookSe ok : BooksNS:BookNeste .phy ;ted ;flat t	at stNested d ( ))) ( ( ))) ( ( ))) ( ( )) ( ) ) ( ) ) ( ) ( ) ( ) ) ( ) ) ( ) ) ( ) ) ( ) ) ( ) ) ) ( ) ) ) ( ) ) ) ( ) ) ) ( ) ) ) )	e bibliography : BooksNS:Bibliography e bookSetMixed : BooksNS:BookSetMixed	< >
		Reset	Keep sorted alphabetica	ly

#### Figure 7.9. Create Virtual Tables From XML Schema Dialog

• Step 4 - Select one or more global schema entities and move from left panel to the right panel. Click **Finish** when done.

When model generation is complete the new model will be opened in an editor for your viewing.

### 7.3. Creating New XML Service Source Model

• To create a new empty xml service source model:

- Step 1 Launch the New Model wizard.
- Step 2 Specify a unique model name.
- Step 3 Select XML Service option from Model Class drop-down menu.
- Step 4 Select Source Model from Model Type drop-down menu.
- Step 5 Click Finish.



### Note

You can change the target location (i.e. project or folder) by selecting the **Browse...** button and selecting a project or folder within your workspace.

- In addition to creating a new empty xml service source model, the following builder option is available:
  - Copy from existing model of the same model class.

### 7.3.1. Copy From Existing Model

This builder option performs a structural copy of the contents of an existing model to a newly defined model. You can choose a full copy or select individual model components for copy.

- To create a new relational model by copying contents from another xml service source model, complete <u>steps 1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Copy from existing model of the same model class and click Next >. The Copy Existing Model dialog will be displayed.
  - Step 6 Select an existing relational model from the workspace using the browse button.



• Step 7 - Check the Copy all descriptions option if desired. Click Finish

New Model Wizard	X
Copy an Existing Model	
Select an existing model, then select model features to be copied.	
Existing Model: Parts.xmi	
Model Contents	
V B PARTS	
A PART_ID : string(4)	
A PART_NAME : string(255)	
A PART_COLOR : string(30)	
PART_WEIGHT : string(255)	≡
SUPPLIER PARTS	~
Copy all descriptions	
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u> Car	ncel

### Figure 7.10. Copy An Existing Model Dialog

### 7.4. Creating New XML Service View Model

- To create a new empty xml service view model:
  - Step 1 Launch the New Model wizard.
  - Step 2 Specify a unique model name.
  - Step 3 Select XML Service option from Model Class drop-down menu.
  - Step 4 Select View Model from Model Type drop-down menu.
  - Step 5 Click Finish.



#### Note

You can change the target location (i.e. project or folder) by selecting the **Browse...** button and selecting a project or folder within your workspace.

- In addition to creating a new empty xml service view model, the following builder option is available:
  - Copy from existing model of the same model class.

### 7.4.1. Copy From Existing Model

This builder option performs a structural copy of the contents of an existing model to a newly defined model. You can choose a full copy or select individual model components for copy.

- To create a new relational model by copying contents from another xml service view model, complete <u>steps 1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Copy from existing model of the same model class and click Next >. The Copy Existing Model dialog will be displayed.
  - Step 6 Select an existing relational model from the workspace using the browse button.



• Step 7 - Check the Copy all descriptions option if desired. Click Finish

New Model Wizard	X
Copy an Existing Model	
Select an existing model, then select model features to be copied.	
Existing Model: Parts.xmi	
-Model Contents	
A PART ID : string(4)	
A PART_NAME : string(255)	
A PART_COLOR : string(30)	
PART_WEIGHT : string(255)	=
P PK_PARTS	
▷ SHIP_VIA	
D STATUS	
SUPPLIER PARTS	<u> </u>
Copy all descriptions	
(?) < Back Next > Finish (	Cancel

Figure 7.11. Copy An Existing Model Dialog

### 7.5. Creating XML Document View Model

- To create a new empty xml document view model:
  - Step 1 Launch the New Model wizard.
  - Step 2 Specify a unique model name.
  - Step 3 Select XML option from Model Class drop-down menu.
  - Step 4 Select View Model from Model Type drop-down menu.
  - Step 5 Click Finish.



#### Note

You can change the target location (i.e. project or folder) by selecting the **Browse...** button and selecting a project or folder within your workspace.

- In addition to creating a new empty xml document view model, the following builder options are available:
  - Copy from existing model of the same model class.
  - Build XML documents from XML schema.

### 7.5.1. Copy From Existing Model

This builder option performs a structural copy of the contents of an existing model to a newly defined model. You can choose a full copy or select individual model components for copy.

- To create a new relational model by copying contents from another xml document view model, complete <u>steps 1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Copy from existing model of the same model class and click Next >. The Copy Existing Model dialog will be displayed.
  - Step 6 Select an existing relational model from the workspace using the browse button.
    - ...
  - Step 7 Check the Copy all descriptions option if desired. Click Finish

New Model Wizard	×
Copy an Existing Model	
Select an existing model, then select model features to be copied.	
Existing Model: Parts.xmi	
Model Contents	
V B PARTS	
A PART_ID : string(4)	
A PART_NAME : string(255)	
A PART_COLOR : string(30)	
PART_WEIGHT : string(255)	
P PK_PARTS	
D SHIP_VIA	
D 🛛 STATUS	
SUPPLIER	
SUPPLIER PARTS	
Copy all descriptions	
	Canad
() < <u>Back</u> <u>Next</u> > <u>Hnish</u>	Cancel

### Figure 7.12. Copy An Existing Model Dialog

### 7.5.2. Build XML Documents From XML Schema

This option creates an XML View document model based on a selected XML schema and its dependencies.

- To create a new xml document view model by from XML schema, complete <u>steps 1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Build XML documents from XML schema and click Next >. The Select XML Schema dialog will be displayed.
  - Step 6 Select an existing schema model from the workspace using the browse button.

....

#### Note

An existing model will be pre-selected if an XSD model in the workspace is selected in the VDB explorer prior to starting the new model wizard. The

schema must be found in the workspace so if you need to get one or more into the workspace use the XSD Schemas on file system importer.

🏧 New Model Wizard			
Select XML Schema Select XML Schema file, then select Schema Root Elements. Or press Fi	ish to create an empty XML Docume	nt.	A
XML Schema File; \Books Project\Books.xsd			
Schema Root Elements	Virtual Documents		
bookCollection : BookSN5:BookSetNested     bookListing : BooksN5:BookSetFlat     bookSetMixed : BooksN5:BookSetMixed	bibliography : BooksN     Solution	S:Bibliography	× >
Document Options  Document Options  Duild full virtual documents from the schema  Duild only first level of each document (for large schema where ma	Reset	└ Keep sorted alphat	betically
Mapping Options     ✓     Build Mapping Classes     Mapping Class Locations	hema-compliant data access		
Mapping Class Column Datatypes © Use XML types from the document © Use "String" mapping type (disables automatic output conversi	n to closest built-in XSD type)		
	< Back	< Next > Finish	Cancel

#### Figure 7.13. Select XML Schema Dialog

• Step 7 - Move the available schema root elements you want to become virtual documents in the new model over to the Virtual Documents list by using the arrow button



for

selected

elements

or



button to move all elements.

• Step 8 - Select the appropriate document options and mapping options. Click Finish

the

• Step 9 - Click Finish to create a model of all selected document entities or (optional) click Next > to view Selected Documents Statistics page which shows document entity statistics and gives you an idea the size of the model being created.

🐢 New Model Wizard	
Selected Documents S	tatistics
This is an overview of the do document, or Finish to creat	ocuments to be generated. Select Next to preview and edit the e the document with default settings.
Documents:	1
Elements:	7
Recursive Elements:	0
Complex Subtype Elements:	0
Attributes:	0
Total entity count:	13
	< Back Next > Finish Cancel

#### Figure 7.14. Selected Documents Statistics Dialog

 Step 10 - (Optional) Click Finish to create a model of all selected document entities or click Next > to view Preview Generated Documents page that allows you to exclude document specific entities then click Finish.

# i Note

For deeply nested schema, your total entity count may be large. If so, displaying the preview may take some time.





### 7.6. Creating XML Schema Model

- To create a new empty xml schema (.xsd) model:
  - Step 1 Launch the New Model wizard.
  - Step 2 Specify a unique model name.
  - Step 3 Select XML Schema (XSD) option from Model Class drop-down menu.
  - Step 4 Select Datatype Model from Model Type drop-down menu.
  - Step 5 Click Finish.



#### Note

You can change the target location (i.e. project or folder) by selecting the **Browse...** button and selecting a project or folder within your workspace.

- In addition to creating a new empty xml schema model, the following builder option is available:
  - Copy from existing model of the same model class.

### 7.6.1. Copy From Existing Model

This builder option performs a structural copy of the contents of an existing model to a newly defined model. You can choose a full copy or select individual model components for copy.

- To create a new relational model by copying contents from another xml schema model, complete steps 1 through 4 above and continue with these additional steps:
  - Step 5 Select the model builder labeled Copy from existing model of the same model class and click Next >. The Copy Existing Model dialog will be displayed.
  - Step 6 Select an existing relational model from the workspace using the browse button.



• Step 7 - Check the Copy all descriptions option if desired. Click Finish

New Model Wizard	X
Copy an Existing Model	
Select an existing model, then select model features to be copied.	
Estation Madels Desta and	
Existing Model: Parts.xmi	
Model Contents	
▼	<u></u>
ART_ID : string(4)	
A PART_NAME : string(255)	
A PARI_COLOR : string(30)	
P PK PARTS	
▶ STATUS	
D SUPPLIER	
SUPPLIER PARTS	~
Copy all descriptions	
(?) < <u>B</u> ack <u>Next</u> > <u>Finish</u>	Cancel

Figure 7.16. Copy An Existing Model Dialog

### 7.7. Creating Web Service View Model

- To create a new empty web service view model:
  - Step 1 Launch the New Model wizard.
  - Step 2 Specify a unique model name.
  - Step 3 Select Web Service option from Model Class drop-down menu.
  - Step 4 Select View Model from Model Type drop-down menu.
  - Step 5 Click Finish.



#### Note

You can change the target location (i.e. project or folder) by selecting the **Browse...** button and selecting a project or folder within your workspace.

- In addition to creating a new empty web service view model, the following builder options are available:
  - Copy from existing model of the same model class.
  - Build from existing WSDL file(s) or URL.

### 7.7.1. Copy From Existing Model

This builder option performs a structural copy of the contents of an existing model to a newly defined model. You can choose a full copy or select individual model components for copy.

- To create a new relational model by copying contents from another web service view model, complete <u>steps 1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Copy from existing model of the same model class and click Next >. The Copy Existing Model dialog will be displayed.
  - Step 6 Select an existing relational model from the workspace using the browse button.



• Step 7 - Check the Copy all descriptions option if desired. Click Finish

6	New Model Wiza	rd 💌
Copy an Existing	Model	
Select an existing mo	del, then select model features to	be copied.
Existing Model: Parts	xmi	
Model Contents		
V B PARTS		
A PART_ID : s	tring(4)	
A PART_NAME	: string(255)	
A PART_COLO	DR : string(30)	
A PART_WEIG	HT : string(255)	
P PK_PARTS		
D SHIP_VIA		
STATUS		
SUPPLIER		
D SUPPLIER PAR	TS	
	ne	
Copy an description	110	
?	< <u>B</u> ack <u>N</u> ext >	<u>Finish</u> Cancel

### Figure 7.17. Copy An Existing Model Dialog

### 7.7.2. Build From Existing WSDL File(s) or URL

This builder option creates a Web service model based on a user-defined WSDL file and its referenced schemas. In addition, applicable XML schema files and XML View document models (optional) are created.

- To create a new relational model by copying contents from another web service view model, complete <u>steps 1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Build from existing WSDL file(s) or URL and click Next >.
  - The remaining wizard steps are identical to those found using the <u>Import > WSDL File into</u> <u>Web Service Model</u> action option.

### 7.7.3. Build From Relational Models

This method is recommended for experienced users for consistent and rapid deployment of **Web** services designed to query relational sources. It provides detailed control of all **Web** service interfaces, operations and required transformations from **XML Views** 

- To create a Web service model from relational models or objects:
  - Step 1 Select any combination of relational models, tables and/or procedures in the Section 4.1, "Model Explorer View" tree..

	ζ	ij	Note It is recommend auto-naming of	ded that th input/out	he user sele put schema	cts sii and \	ngle sour Veb serv	ce mod	els, which e lels in \ <b>Ste</b>	enables <b>p 3</b> .
• {	Step	2	- Right-click	select	Modeling	>	Create	Web	Service	action
	e	Model I	Explorer 🗙	Outline				- 🗆		
				$a_z \otimes$	$\langle \Rightarrow \Rightarrow  $	2		$\overline{}$		
	E-1	🛃 Allo	CustomerAccour	nts						
		÷	CustomerWS.>	mi						
		÷ 🔽	CustomerWSR	esponse:	s.×mi					
			Northwind_Or/	acle, xmi l	2	-				
		+	Northwind_V	New	-1.1.1					
				New (		1				
				Creat	e Web Serv	/ice				
				82						

### Figure 7.18. Create Web Service Action

• Step 3 - In the Create Web Service dialog, specify file names for the generated Input Schema file, Output Schema file and Web service model. Change options as desired. Click Finish when done.

🙅 Create Web Servic	e
Create Web Service Click Finish to generate.	
To generate a Web Servi	ce Model an Output Schema, Output XML View and an Input Schema are required.
Output Options:	
Generate Output Sch	ema
🔽 Generate flat XSD	
Output Schema Name:	Northwind_Oracle_Output
Generate XML Views I	or Output Schema
🔽 Generate default SQL	. for XML mappings
Input Options:	
Generate Input Scher	na
Input Schema Name:	Northwind_Oracle_Input
Web Service Options:	
Generate Web Servic	e Model
Web Service Model Name	Northwind Oracle WS
1.00.00.110.110	
	Finish Cancel

### Figure 7.19. Create Web Service Dialog

• Step 4 - When model generation is complete, a confirmation dialog should appear. Click OK.

Generation completed successfully.	
Generation completed successfully.	
New Model Northwind_Oracle_Output.xsd created New Model Northwind_Oracle_Input.xsd created New Model Northwind_Oracle_Output_View.xmi created New Model Northwind_Oracle_WS.xmi created	
	ОК

### Figure 7.20. Generation Completed Dialog

### 7.7.4. Build From XML Document View Models

Web Service models and their corresponding **Interfaces** and **Operations** can be generated in Teiid Designer from **XML View** model components. Namely, XML View Documents and XML View Document roots.

- To create a new Web service model from XML components::
  - Step 1 Select either a single XML Document or single XML Document root in Section 4.1, "Model Explorer View".
  - Step 2 Right-click select Modeling > Create Web Service action
     Image: Im
  - Step 3 Fill in missing properties in Web Service Generation Wizard shown below.

DDCC	ify location for gene	arated mod	lel(s) (i.e. Project or Folder)	
AllCu	ustomerAccounts		Brow	se
Bold	Blue text indicates	an existing	g workspace entity. Bold Red text indicates an error.	
1	WebService Model:	Northwir	nd Oracle WS	
I	Interface Name:	REGION	Output	
¢	Operation Name:	getREGI	ON Output	
-Of	peration Definit	tion	CUSTOMERS_Input : CUSTOMERS_Input_Type .	]
	Input Message N	Jame:	REGION_Input	
	Output Message	Element:	e REGION_Output	

### Figure 7.21. Generate A Web Service Dialog

• Step 4 - Click Finish to generate model. When model generation is complete, a confirmation dialog should appear. Click OK.



Figure 7.22. Generation Completed Dialog



#### Note

Users can change the **Web Service Model** and **Interface Name** values (via "..." buttons) to use existing **Web service** model components. This will create a new operation in an existing model.

### 7.8. Creating New Extensions Model

- To create a new empty extensions model:
  - Step 1 Launch the New Model wizard.
  - Step 2 Specify a unique model name.
  - Step 3 Select Model Extensions option from Model Class drop-down menu.
  - Step 4 Select Model Class Extension from Model Type drop-down menu.
  - Step 5 Click Finish.



#### Note

You can change the target location (i.e. project or folder) by selecting the **Browse...** button and selecting a project or folder within your workspace.

- In addition to creating a new empty extensions model, the following builder option is available:
  - · Copy from existing model of the same model class.

### 7.8.1. Copy From Existing Model

This builder option performs a structural copy of the contents of an existing model to a newly defined model. You can choose a full copy or select individual model components for copy.

- To create a new relational model by copying contents from another extensions model, complete <u>steps 1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Copy from existing model of the same model class and click Next >. The Copy Existing Model dialog will be displayed.
  - Step 6 Select an existing relational model from the workspace using the browse button.
  - Step 7 Check the Copy all descriptions option if desired. Click Finish

New Model Wizard	X
Copy an Existing Model	
Select an existing model, then select model features to be copied.	
Existing Model: Parts.xmi	
Model Contents	
V B PARTS	
A PART_ID : string(4)	
A PART_NAME : string(255)	
A PART_COLOR : string(30)	
PART_WEIGHT : string(255)	≡
SUPPLIER PARTS	~
Copy all descriptions	
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u> Car	ncel

### Figure 7.23. Copy An Existing Model Dialog

### 7.9. Creating New Relationship Model

- To create a new empty relationships model:
  - Step 1 Launch the New Model wizard.
  - Step 2 Specify a unique model name.
  - Step 3 Select Relationship option from Model Class drop-down menu.
  - Step 4 Select Logical Model from Model Type drop-down menu.
  - Step 5 Click Finish.



#### Note

You can change the target location (i.e. project or folder) by selecting the **Browse...** button and selecting a project or folder within your workspace.

- In addition to creating a new empty relationship model, the following builder option is available:
  - Copy from existing model of the same model class.

### 7.9.1. Copy From Existing Model

This builder option performs a structural copy of the contents of an existing model to a newly defined model. You can choose a full copy or select individual model components for copy.

- To create a new relational model by copying contents from another relationships model, complete <u>steps 1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Copy from existing model of the same model class and click Next >. The Copy Existing Model dialog will be displayed.
  - Step 6 Select an existing relational model from the workspace using the browse button.



- Step 7a If Copy entire model option is selected, Click Finish
- Step 7b If Select Model Features to Copy option is selected, select and/or de-select desired model objects to be included or excluded from the structural copy and click Finish.
| New Model Wizard   | X    |
|--|------|
| Copy an Existing Model   |      |
| Select an existing model, then select model features to be copied. |      |
|  |      |
|  |      |
| Existing Model: Parts.xmi  |      |
| Model Contents   |      |
| V B PARTS  |      |
| A PART_ID : string(4)  |      |
| A PART_NAME : string(255)  |      |
| A PART_COLOR : string(30)  |      |
| PART_WEIGHT : string(255)  | ≡    |
|  |      |
|  |      |
|  |      |
| SUPPLIER PARTS   | ~    |
|  |      |
| Copy all descriptions  |      |
|  |      |
|  |      |
|  |      |
| (?) < <u>Back</u> <u>Next</u> > <u>Finish</u> Car                  | ncel |

## Figure 7.24. Import Wizard Selection Dialog

## 7.10. Creating New UML 2.0 Model

- To create a new empty UML 2.0 model:
  - Step 1 Launch the New Model wizard.
  - Step 2 Specify a unique model name.
  - Step 3 Select UML 2.0 option from Model Class drop-down menu.
  - Step 4 Select Logical Model from Model Type drop-down menu.
  - Step 5 Click Finish.



#### Note

You can change the target location (i.e. project or folder) by selecting the **Browse...** button and selecting a project or folder within your workspace.

- In addition to creating a new empty uml model, the following builder option is available:
  - Copy from existing model of the same model class.

## 7.10.1. Copy From Existing Model

This builder option performs a structural copy of the contents of an existing model to a newly defined model. You can choose a full copy or select individual model components for copy.

- To create a new relational model by copying contents from another UML model, complete <u>steps</u> <u>1 through 4</u> above and continue with these additional steps:
  - Step 5 Select the model builder labeled Copy from existing model of the same model class and click Next >. The Copy Existing Model dialog will be displayed.
  - Step 6 Select an existing relational model from the workspace using the browse button.



• Step 7 - Check the Copy all descriptions option if desired. Click Finish

New Model Wizard				
Copy an Existing Model				
Select an existing model, then select model features to be copied.				
Existing Model: Parts.xmi				
Model Contents				
V B PARTS		<u>^</u>		
A PART_ID : string(4)				
PART_NAME : string	g(255)			
A PART_COLOR : stri	ng(30)			
PART_WEIGHT : str	ing(255)	=		
P PK_PARTS				
		~		
Copy all descriptions				
?	< Back Next > Finis	h Cancel		

Figure 7.25. Copy An Existing Model Dialog

# **Creating and Editing Model Objects**

#### Go To: Table of Contents [index.html]

This section summarizes Teiid Designer features for creating and editing existing model objects contained in your models.

## 8.1. Creating New Model Objects

As discussed in the introduction, *Metadata Models* provide a framework to model various types of metadata. Each metamodel type has a set of parent-child relationships that establish constraints on what can be created and where. You cannot, for example, create a column attribute in a stored procedure, nor can you create a mapping class column in a Web service operation's output message.

The Teiid Designer provides a common set of actions to create new children of these models as well as children of children.

- You can create new model objects directly in the Section 4.1, "Model Explorer View" view, Diagram Editor or Table Editor using the following actions:
  - New Child Action
  - New Sibling Action
  - New Association Action

## 8.1.1. New Child Action

- To create new child model objects in the Section 4.1, "Model Explorer View":
  - **Step 1** Select the parent object to which you want to add a child. For example, you can add a package to a package or an attribute to a class.
  - Step 2 Right-click on a container object. From the pop-up menu, select New Child. You can now select the child object you would like to add.

Model Explorer 🗙	Outline 🗖 [		
and the second se	Iª 🕸 ⇔ ⇔ @   🖻 😫 '		
🖻 📬 Northwind_	Oracle.xmi	^	
⊡ 1≣ import o ™ <mark>™</mark> Package	eclarations (2) 9 Diagram		
	New Child	Þ	Access Pattern
	New Sibling New Association	•	Column
	Create Web Service		F Foreign Key P Primary Key
	🖓 Undo		🔟 Unique Constraint
	🍄 Redo		

## Figure 8.1. New Child Action In Model Explorer

• Step 3 - The new model object displays on the Section 4.1, "Model Explorer View" and is highlighted for renaming.

Model Explorer 🖾 Outline	- 0
📭 🤣 🗇 🖓 🗐 🔁	< <
🖻 🙀 Northwind_Oracle.xmi	~
- 🔠 Package Diagram	
CATEGORYID : long	-
CATEGORYNAME : string(30)	
DESCRIPTION : string(4000)	
A NewColumn	
PICTURE : blob	
PK_CATEGORIES	

### Figure 8.2. New Model Object In Explorer

- To create new child model objects in the *Diagram Editor*:
  - **Step 1** Select the parent object to which you want to add a child. For example, you can add a package to a package or an attribute to a class.
  - Step 2 Right-click on a container object. From the pop-up menu, select New Child. You can now select the child object you would like to add.

≪⊲Base Table>> ≓ SHIDDEDS		
	New Child	📕 🔺 Access Pattern
SHIPPERID : long     A COMPANYNAME : string     A PHONE : string(48)	New Sibling New Association	
	Create Web Service	Primary Key
	💛 Undo Create NewColumn 铃 Redo	Unique Constraint

## Figure 8.3. New Child Action In Diagram

• Step 3 - The new model object displays on the diagram and is highlighted for renaming.



Figure 8.4. New Model Object In Diagram

- To create new child model objects in the *Table Editor*:
  - **Step 1** Select the row for the parent object to which you want to add a child. For example to add a column, click the **Base Table** tab and select base table row.
  - Step 2 Right-click on a table row. From the pop-up menu, select New Child. You can now select the child object you would like to add.

诸 *Northwind	1_Oracle.xmi 🗙				
🗾 Base Ta	bles 🖪 Column	s 🖪 Foreign Keys 🔳 Indexes	P Primary Keys		
Location	Name	Name In Source		System	Card
	CATEGORIES	CATEGORIES		false	7
	CUSTOMER	New Child	👌 📐 📐	ess Pattern	1
	CUSTOMER	New Sibling	🚺 🖾 Colu	mn	
	EMPLOYEE	New Association	F Fore	ign Key	
		🕗 Undo Create NewColumn	P Prim	ary Key	
		> Redo	🗾 Uniq	ue Constraint	37
	REGION	t ot		false	4
	SHIPPERS	j cu		false	3

### Figure 8.5. New Child Action In Table Editor

• Step 3 - The selected tab in the Table Editor changes to the tab for the child object type, the new model object row is displayed and the row's name table cell is highlighted for renaming.

🗾 Base Table	s 🔼 Columns	E Foreign Ke	eys
Location	Name	Name In So	Nativ
SUPPLIERS	COUNTRY	COUNTRY	VARG
SUPPLIERS	PHONE	PHONE	VARG
SUPPLIERS	FAX	FAX	VARC
SUPPLIERS	HOMEPAGE	HOMEPAGE	VARC
TERRITORIES	TERRITORYID	TERRITORYID	VARC
TERRITORIES	TERRITOR	TERRITOR	CHAP
TERRITORIES	REGIONID	REGIONID	NUM
SHIPPERS	NewColumn		
CATEGORIES	NewColumn1		
1		100	

## Figure 8.6. New Model Object In Table Editor

## 8.1.2. New Sibling Action

- To create new sibling model objects in the Section 4.1, "Model Explorer View":
  - **Step 1** Select the object to which you want to add a sibling. For example, you can add a column sibling to a column.

• Step 2 - Right-click on that object. From the pop-up menu, select New Sibling. You can now select the sibling object you would like to add.



## Figure 8.7. New Sibling Action In Model Explorer

• Step 3 - The new model object displays on the Section 4.1, "Model Explorer View" and is highlighted for renaming.



## Figure 8.8. New Model Sibling In Explorer

• To create new sibling model objects in the *Diagram Editor*:

- **Step 1** Select the object to which you want to add a sibling. For example, you can add a column sibling to a column.
- Step 2 Right-click on that object. From the pop-up menu, select New Sibling. You can now select the sibling object you would like to add.

📽 *Northwind_Oracle.xmi 🗙		
	Ress ERS www.Child www.Sibling www.Association ndo Delete multiple objects Ctrl+Z edo	<ul> <li>Access Pattern</li> <li>Column</li> <li>Foreign Key</li> <li>Primary Key</li> <li>Unique Constraint</li> </ul>

## Figure 8.9. New Sibling Action In Diagram

• Step 3 - The new model object displays on the diagram and is highlighted for renaming.



### Figure 8.10. New Sibling Object In Diagram

- To create new sibling model objects in the *Table Editor*.
  - **Step 1** Select the row for the object to which you want to add a sibling. For example, you can add a column sibling to a column.

• Step 2 - Right-click on a row. From the pop-up menu, select New Sibling. You can now select the sibling object you would like to add.

*Northwind Base Tal	l_Oracle.xmi × bles A Colur	nns 🖪 Foreign Keys 🔳 Indexe	es Prima	ry Keys	
Location	Name	Name In Source		System	Cardin
	CATEGO CUSTOM	New Child	•	false Falsa	7
	CUSTOM CUSTOM EMPLOYI	New Sibling New Association	•	Base Table	
	ORDERS	Undo Create NewColumn1	1	] Index	7
		Redo Cut	[	<ul> <li>Logical Relationship</li> <li>Procedure</li> </ul>	37
	SHIPPER SUPPLIE	Сору		Schema View	
	TERRITO	Table Paste	21	Custom Diagram	
		Copy Name			

## Figure 8.11. New Sibling Action In Table Editor

• Step 3 - The selected tab in the Table Editor changes to the tab for the child object type, the new model object row is displayed and the row's name table cell is highlighted for renaming.

🗾 Base Table	s 🔼 Columns	E Foreign Ke	eys   [
Location	Name	Name In So	Nativ
SUPPLIERS	COUNTRY	COUNTRY	VARG
SUPPLIERS	PHONE	PHONE	VARC
SUPPLIERS	FAX	FAX	VARC
SUPPLIERS	HOMEPAGE	HOMEPAGE	VARC
TERRITORIES	TERRITORYID	TERRITORYID	VARC
TERRITORIES	TERRITOR	TERRITOR	CHAP
TERRITORIES	REGIONID	REGIONID	NUM
SHIPPERS	NewColumn		
CATEGORIES	NewColumn1		
<		101	

Figure 8.12. New Sibling Object In Table Editor

## 8.1.3. New Association Action

- To create new associations between model objects in the Section 4.1, "Model Explorer View":
  - **Step 1** Select two objects you wish to associate. For example, select columns in different base tables.
  - Step 2 Right-click. From the pop-up menu, select New Association > Foreign Key Relationship..



## Figure 8.13. New Association Action In Model Explorer

• Step 3 - The new relationship link is displayed in the diagram.

1* M	lorthwind_Oracle.xmi ×	
	< <base table=""/> > REGION REGIONID : long REGIONDESCRIPTION : string(100) NewForeignKey	< <base table=""/> >         Image: Shippers         Image: Shippers

Figure 8.14. New Association In Diagram

- To create new associations between model objects in the *Diagram Editor*.
  - **Step 1** Select two objects you wish to associate. For example, select columns in different base tables.
  - Step 2 Right-click. From the pop-up menu, select New Association > Foreign Key Relationship..



### Figure 8.15. New Association Action In Model Explorer

• **Step 3** - The new relationship link is displayed in the diagram. The Column, Foreign Key, Primary Key reference properties are properly set on the selected columns, new primary key and new foreign key.

👫 *Nort	hwind_Oracle.xmi ×	
	< <base table=""/> >	< <base table=""/> > SHIPPERS SHIPPERID : long COMPANYNAME : string(80) PHONE : string(48) PK_SHIPPERS

## Figure 8.16. New Association In Diagram

OR

- Step 1 Select a column in table.
- **Step 2** Drag the column to another table and drag over a column and drop onto this column. The target column should highlight in Yellow.

Image: Second secon	< <base table=""/> > SHIPPERS COMPANYNAME : string(80) PHONE : string(48)
--	--

## Figure 8.17. New Association Via Drag-and-Drop In Diagram

• **Step 3** - The new relationship link is displayed in the diagram. The Column, Foreign Key, Primary Key reference properties are properly set on the selected columns, new primary key and new foreign key.

💏 *No	rthwind_Oracle.xmi ×	
	< <base table=""/> >   REGION   REGIONID : long   REGIONDESCRIPTION : string(100)	< <base table=""/> > SHIPPERS  SHIPPERID : long COMPANYNAME : string(80) PHONE : string(48)  PL PK_SHIPPERS

Figure 8.18. New Association In Diagram

- To create new associations between model objects in the *Table Editor*.
  - **Step 1** Select two objects you wish to associate. For example, select columns in different base tables.
  - Step 2 Right-click. From the pop-up menu, select New Association > Foreign Key Relationship.

💏 *Northwind_O	racle.xmi 🗙							
🗾 Base Tables	; 🖪 Columns 🖪	E Foreign k	(eys 🛛 🗓 Index	ces 📔 🖻 Primar	ry Key	s		
Location	Name		Name In So	Native Type	Len	igth	Length Fixed	Numeri
CATEGORIES	CATEGORYID		CATEGORYID	NUMBER	0		true	10
CATEGORIES	CATEGORYNAME		CATEGORY	VARCHAR2	30		false	0
CATEGORIES	DESCRIPTION		DESCRIPTION	VARCHAR2	4000		false	0
CATEGORIES	PICTURE		PICTURE	BLOB	0		false	0
CUSTOMERC	CUSTOMERID		CUSTOMEDID	Гсилп	119		true	0
CUSTOMERC	CUSTOMERTYP	New Child					true	0
CUSTOMERD	CUSTOMERTYP	New Siblin	na		+1		true	0
CUSTOMERD CUSTOMERS	CUSTOMERDES	New Association				L. Relatio	onship	
CUSTOMERS		A		12	F	Eoreia	n Key Relationshin	
CUSTOMERS		Undo Dele	ete multiple objec	ts		_ r or org		
CUSTOMERS	CONTACTTITLE 😂	Redo			1		false	0
CUSTOMERS	ADDRESS -	1. 19922-92				)	false	0

## Figure 8.19. New Association Action In Model Explorer

• Step 3 - New Foreign Key and Primary Key objects will be added to the contents of their respective tabs in the Table Editor. The Column, Foreign Key, Primary Key reference properties are properly set on the selected columns, new primary key and new foreign key.

## 8.2. Model Object Editors

Teiid Designer provides specialized object editors to handle complex model objects and their unique properties. These objects include:

- <u>Transformations</u>
- Mapping Class Input Sets
- XML schema choice elements
- <u>Recursive Mapping Classes</u>
- <u>Web Service Operations</u>

This section describes these editors in detail.

In addition, the primary actions for editing model objects are:

• • t

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.

- **Clone** Duplicates the selected object in the same location with the same name; user is able to rename the new object right in the tree.
- ×

Delete - Deletes the selected object(s).

• Rename - Allows a user to rename an object.

These actions are presented in Teild Designer's main **Edit** menu and also in the right-click context menus for model objects selected in the *Model Explorer*, *Diagram Editor* and *Table Editor*.

## 8.2.1. Transformation Editor

The **Teiid Designer's Transformation Editor** enables you to create the query transformations that describe how to derive your virtual metadata information from physical metadata sources or other virtual metadata and how to update the sources.

The **Transformation Editor** provides a robust set of tools you can use to create these SQL queries. You can use these tools, or you can simply type a SQL query into the Transformation Editor.

To edit a transformation you can:

- Double-click Edit
  - A relational view table or procedure in the Model Explorer or Diagram Editor
  - A transformation node in a transformation diagram or mapping transformation diagram
  - A mapping class in a mapping diagram or mapping transformation diagram
- Right-click Edit action on selected object in the Model Explorer, Diagram Editor or Table Editor
  - A relational view table or procedure
  - A transformation node in a transformation diagram or mapping transformation diagram
  - A mapping class in a mapping diagram or mapping transformation diagram

If a Model Editor is not currently open for the selected object's model, a Model Editor will be opened.

After the corresponding transformation diagram is opened in the Diagram Editor, the Transformation Editor is displayed in the lower section of the Diagram Editor.



### Figure 8.20. Editing String Property

If this virtual class supports updates, the tabs on the bottom of the **Transformation Editor** allow you to enter SQL for each type of query this virtual class supports. If this virtual class does not support updates, only the **SELECT** tab is available.

You can enter separate SQL queries on each available tab to accommodate that type of query.

Within the Transformation Editor, you can:

- Disable specific update transformation types on this virtual class.
- Build or edit a criteria clause to use in your transformation.
- Build or edit an expression to use in your transformation.
- Find and replace a string within your transformation.

- Validate the transformation to ensure its content contains no errors.
- Reconcile target attributes to ensure the symbols in your transformation match the attributes in your virtual metadata class.

You can also set preferences that impact the display of your **Transformation Editor**. For more information, see Transformation Editor Preferences.

• The Transformation Editor toolbar actions are summarized below.



**Prevew 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. Simply 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 checked, the editor shows four tabs at the bottom for the Select, Update, Insert and Delete transformations. If 'Supports Update' is unchecked, all updates are disabled and only the Select transformation is displayed.

• 🗖 גַּ

**Reconcile** - allows you to resolve any discrepancies between the transformation symbols and the target attributes. Pressing this button will display the "Reconcile Virtual Target Attributes" dialog box in which you can resolve discrepancies.

## · v<sub>ə</sub>

**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.

## ۰ A

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.

#### • a.b.c + c

**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 a 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" - closes the transformation editor.

## 8.2.1.1. Using the Criteria Builder

The Transformation Editor's **Criteria Builder** offers you a quick, graphical means to build criteria clauses in your transformations based on meta objects in your diagram. If you launch the **Criteria Builder** with your cursor within an existing criteria in your transformation SQL, the builder will open in Edit mode. If your cursor is not in an existing criteria location, the builder will open in create mode and allow you to create it from scratch.

This procedure provides an example of building a criteria clause using the **Criteria Builder**. When building your own criteria, you can mix and match the values and constants with whatever logic you need to build powerful and complex criteria.

#### • To use the Criteria Builder:

- Step 1 In the Transformation Editor, click the Launch Criteria Builder button.
- Step 2 The Criteria Builder displays.



Figure 8.21. Editing String Property

The two tabs at the top, **Tree View** and **SQL View**, show the current contents of the criteria you have built.

The **Criteria Editor** at the bottom allows you to build a criteria clause. To build a criteria clause, you must add information to the left side of the predicate, select a comparison operator, and add a value to the right side.

- Step 3 The radio buttons on either side of the **Predicate Editor** let you choose what type of content to place in that side of your predicate. Click the radio button of the type of content you want to place in your criteria. You can click:
  - Attribute to add an attribute to the predicate. If you click the Attribute radio button, the Predicate Editor looks like this:



#### Figure 8.22. Attribute Panel

From the tree, select the attribute you want to add to the expression. You can select an attribute from any of the source classes in the transformation.

• **Constant** to add a hard-wired constant value to the predicate. If you click this radio button, the **Predicate Editor** looks like this:

CAttribute 🖲 Constant C Function	
Apply' to replace selected Attribute with Constant below.	
ype: string	•
Value	
024A-X234	

### Figure 8.23. Constants Panel

Select the datatype for this constant from the Type drop-down list and enter the value in the Value edit box.

• **Function** to add a function. If you click the Function radio button, the Predicate Editor looks like this:



Figure 8.24. Functions

Click the Edit button to use the Expression Builder to construct a function to use in the predicate of your SQL Criterion. For more information about the Expression Builder, see "Using the Expression Builder."

- Step 4 Set a value left side of the predicate and, when necessary, the right side of the predicate. If the right side of the predicate does not require a value of some sort, the Criteria Builder will not let you enter one.
- Step 5 Click Apply.
- **Step 6** When you have created both a Left Expression and a Right Expression in the Predicate Editor, click Apply to add the criterion to the tree view at the top of the dialog box.

The criteria clause displays in the Criteria tree.

You can create complex criteria by joining other criteria with this one. To join criteria with this one, select the criteria in the Criteria tree and click:

- Delete to remove the selected criterion.
- AND to create a new criterion that must also be true.
- OR to create a new criterion that can be true instead of the selected criterion.
- NOT to establish negative criterion.

If you join a criterion to the one you just completed, you build the expression the same way, using the Expression Editors panel and the Predicate Editor panel. You can create complex, nested criteria by judicious use of the AND and OR buttons.

Once you have created the complete criteria you want, click OK to add it to your transformation.

## 8.2.1.2. Using the Expression Builder

The **Transformation Editor's Expression Builder** offers you a quick, graphical means to build expressions in your transformations. This **Expression Builder** lets you create:

- Attributes by selecting an attribute.
- Constants by selecting the datatype and value.
- Functions from both the standard Teiid Designer SQL functions and your enterprise's custom user-defined functions. If you select a function before you launch the Expression Builder, you can use the Expression Builder to edit the selected function; otherwise, you can create a new function from scratch.
- To use the Expression Builder:
  - Step 1 In the Transformation Editor, click the location where you want to insert the function.

• Step 2 - Click the Expression Builder button. The SQL Expression Builder displays.



#### Figure 8.25. Expression Builder

The two tabs at the top, Tree View and SQL View, show the current contents of the expression you have built. To build an expression, you must specify the type of expression you want to build and populate it. In most cases, you will use the Expression Builder to construct a complex expression.

• Step 3 -Click the Function radio button to add a function.



• Step 4 - The Expression Editor displays the Function editor.

Expression Bu	uilder		
nitial Expression: L	ASTNAME		
Tree View SQL Vie	w		
(⑦ <undefine< th=""><th>d&gt;</th><th></th><th>Delete</th></undefine<>	d>		Delete
		Expression Editors	
C Attribute Apply' to replace Category: STI Function: CO Argument Nar STRING1 STRING2	Constant Fu te selected <undefin RING WCAT(STRING1, STI me Value <undefined> <undefined></undefined></undefined></undefin 	Inction hed > with Function below.	Apply Reset
		OK	Cancel

#### Figure 8.26. Function Panel Selected

From the Category drop-down list, choose the type of function you want to add. By default, the Teiid Designer System offers the following categories:

- **Conversion** for functions that convert one datatype into another.
- **Datetime** for functions that handle date or time information.
- Miscellaneous for other functions.
- Numeric for mathematic and other numeric functions.
- String for string manipulation functions.



#### Note

Any additional categories represent those containing user-defined functions your site has created.

- **Step 5** From the **Function** drop-down list, select the function you want. The table beneath the drop-down lists displays the number of arguments required for this function.
- Step 6 Click Apply.
- **Step 7** Your function displays in the tree at the top. Sub nodes display for each argument you need to set for this function.

🐢 Expression Builder	
Initial Expression: LASTNAME	
Tree View SQL View	
CONCAT( <undefined>, <undefined>) ③ <undefined> ④ <undefined> ④ <undefined></undefined></undefined></undefined></undefined></undefined>	Delete
Expression Editors	
Attribute Constant Function      Attribute Constant Function      Apply' to replace selected <undefined> with Attribute below.      Attribute below.      Books_SourceB.BOOK_AUTHORS      Books_SourceB.BOOK_AUTHORS.ISBN      Books_SourceB.BOOK_AUTHORS.AUTHOR_ID      MINPUT.isbn      Books_SourceA.AUTHORS.AUTHOR_ID      Books_SourceA.AUTHORS.AUTHOR_ID      Books_SourceA.AUTHORS.FIRSTNAME      Books_SourceA.AUTHORS.LASTNAME      Books_SourceA.AUTHORS.MIDDLEINIT</undefined>	Apply Reset
OK	Cancel

#### Figure 8.27. New Blank Function Created

You need to set an attribute or constant value for each sub node in the tree to specify the arguments this function needs. You can also nest another function in the tree using the **Function** editor.

🙅 Expression Bui	lder		×				
Initial Expression: LA	STNAME						
Tree View SQL View							
□       fx       CONCAT( <undefined>, <undefined>),         □       fx       CONCAT(<undefined>, <undefined>)         □       fx       cundefined&gt;         ○       <undefined>       □</undefined></undefined></undefined></undefined></undefined></undefined></undefined></undefined></undefined></undefined></undefined></undefined></undefined>							
		Expression Editors					
C Attribute C Choose Function Category: STR Function: CON Argument Name	Constant  Fu below to replace so ING CAT(STRING1, STR Value	nction elected <undefined>. RING2)</undefined>	Apply				
STRING1 STRING2	<undefined> <undefined></undefined></undefined>						
		ОК	Cancel				

## Figure 8.28. Nested Function Example

- **Step 8** Click each sub node in the tree and use the editors at the bottom of the dialog box to apply an attribute, constant, or function value to it.
- Step 9 When you have added values to all nodes, as shown below, click OK. to add this expression to your query or Cancel to close the dialog box without inserting the expression.

If the **OK** button does not enable, you have not added a value to all nodes in the tree.



### Figure 8.29. Completed Expression

You can also nest functions within your expressions by selecting an argument and selecting a function for that argument. The nested function displays in the tree beneath your root function and its arguments display as well. Using the Expression Builder and nested functions, you can create complex logic within your query transformations.

## 8.2.2. Input Set Editor (XML)

The **Input Set** represents a special class that contains attributes from a parent mapping class. When you create mapping classes for an **XML Document** model, the Teiid Designer automatically adds an **Input Set** to all XML transformation diagrams for mapping classes beneath the highest node in the Document meta object.

The **Input Set** proves especially useful for information integration using the **Teiid Designer Server**. Through the **Input Set**, you can access a row of data generated by any XML transformation in a mapping class higher in the XML document's hierarchy. You can use **Input Set** attributes, which are individual columns from the rows of data, within the criteria of an XML transformation query of the child mapping class. You cannot use the **Input Set** attributes within the SELECT portion of the XML transformation query.

To use an Input Set, you must use the Input Set Editor to bind attributes from parent classes.

Once you have created an **Input Set**, you can use the attributes within it as source material for the XML transformation diagram's query.

The **Input Set** only serves to enable data flow between nested mapping classes. If you use the **Teiid Designer Server** for data access, your applications cannot directly query an **Input Set**. **Input Sets** only display in the XML transformation diagram to which they belong. **Input Sets** do not display on the *Section 4.1, "Model Explorer View*" view and you cannot use them as you would a normal class, such as for source classes in other transformations.

To open the **Input Set Editor**, either double-click the input set in the **Mapping Transformation Diagram** or click the edit button on the **Input Set** in the diagram. (see below)



Figure 8.30. Edit Input Set Button

Input Parameters	Mapping Class Bindings	🖃 📰 book	
🛛 isbn : string	A book.isbn : string	< New isbn : string	
		< Bind Cype : string	
		Unbind > A publisher : string	g
		Delete	

### Figure 8.31. Input Set Editor Panel

The **Input Parameters** table contains a list of mapping attributes within the input set and the mapping attributes bound to input set mapping attributes. The tree on the right displays the parent mapping classes and the attributes available from each.

Using the Input Set Editor, you can:

- Add a mapping attribute from a parent mapping class to the **Input Set**. In the tree on the right, select the symbol for which you want to create an attribute and click **New**. The item displays in the **Input Parameters** and **Mapping Class Bindings** table.
- Delete a mapping attribute from the Input Set. Click the row in the Input Parameters and Mapping Class Bindings table that you want to delete and click Delete. The Teiid Designer removes this row from the table and this mapping attribute from your Input Set.
- Bind and Unbind Input Parameters.

Once you have created the mapping attributes within the **Input Set** that you need, you can use the **Input Set Parameters** within a mapping class transformation to produce mapping attributes you can map to your XML document.

Input Parameters	Mapping Class Bindings	Dook
🕅 isbn : string	book.isbn : string	< New isbn : string < Bind type : string < Bind title : string
		Unbind > A publishDate : string Delete

## Figure 8.32. Input Set Editor Panel

## 8.2.3. Choice Editor (XML)

Within an XML Document model, a choice compositor defines all possible document data structures (sometimes called fragments) that can appear at that location in an XML instance document. When the Teiid Designer Server populates an XML instance document at runtime based upon your virtual XML document, it will choose the first fragment that matches the criteria you specify within the **Choice Editor**.

To view the choice editor, right-click on the choice node in the mapping diagram's XML Document tree view and select **Edit** from the right-click pop up menu.



Figure 8.33. Opening The Choice Editor

	Option Name	Include	Criteria
1	😟 hardcoverBook		AudioBooks.bookSetMixedDocument.MappingClasses.bookSetMixed.TYPE = 'Hardback'
2	(0) softcoverBook	~	UCASE(AudioBooks.bookSetMixedDocument.MappingClasses.bookSetMixed.TYPE) LI
3	(e) audioBook	$\checkmark$	AudioBooks.bookSetMixedDocument.MappingClasses.bookSetMixed.TYPE IN ('Books

## Figure 8.34. The Choice Editor

The table on this panel displays fragment options for the choice, each represented by the top node of the document fragment.

The Summary tab, shown below, displays a SQL-like version of the current choice criteria.



### Figure 8.35. Choice Criteria Summary

## 8.2.3.1. Using the Choice Editor

You should address each choice option by performing one of the following:

- **Specify a criteria** statement for the Teiid Designer Server to apply in order to determine which elements or elements to insert into the result document.
- Exclude or include the option's fragment from the document.
- Set the elements' criteria test order.
- Set a default action that occurs if none of the criteria you set is met.

### 8.2.3.2. Excluding Fragments

The XML Schema upon which you based the XML Document model determines the nature of the options available to the choice. A schema you share with other, external sources (such as business partners) might include information that you do not want to include within XML files.

For example, Sample Financial Services shares an XML schema with its partners Example Mutual Insurance, Illustrative Brokerage, and FinancialPartners.com. The partners created the schema

broadly, to cover all possibilities for information they might need to interchange. As such, the customer information XML document might include a choice compositor based on a list of all products all companies offer.

However, Sample Financial does not offer a credit card; so it could exclude those elements from the XML documents its Teiid Designer Server creates since it will never have credit card information for an XML document.

The table on the **Choice Editor** contains the **Include** column. By default, all elements specified by the schema are included. You can click to remove the checkmark beside any element you do not want to include within your XML documents generated by this virtual XML document metadata model. By removing the checkmark, you are not removing the element from the XML Document model; you are merely telling the Teiid Designer Server that it will never use this element as part of the choice.

You cannot edit criteria for excluded elements. However, if you exclude an option for which you have established a criteria, Teiid Designer will retain the criteria if you want to include the option in the future.

## 8.2.3.3. Editing Choice Criteria

- To edit the criteria for a choice element:
  - Step 1 In the table on the Choice Editor panel, select the element you want to edit..
  - Step 2 Click Edit Criteria button to launch the Criteria Builder dialog.
  - Step 3 Use the Criteria Builder to create the conditions for which the Teiid Designer Server will test to determine whether to choose this option in the XML instance document.
  - Step 4 Click OK. The criteria you set displays both in the table and in the summary tab.

You must set a criterion for each option in your document unless you have selected to exclude that option or specify that option will be the default option.

## 8.2.3.4. Setting Choice Element Order

To edit the criteria for a choice element:

The Teiid Designer Server evaluates the choice criteria in the order in which they appear, and when one choice criteria is met, the Teiid Designer Server populates the XML instance document with that option. The Teiid Designer Server might not test all criteria for all options, so their order matters a great deal.

Therefore, the order in which your options appear within the choice criteria often determines what information appears ultimately in your XML instance documents. You can reorder the option list within the choice to set the order in which the Teiid Designer Server tests the criteria.

9

То	set	this	order,	select	an	element	in	the	table	and	use	the
1992												
	0											
212222	5223											
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.

## 8.2.3.5. Setting a Default Choice Action

The default action represents the course the Teiid Designer Server should take if none of the criteria you set evaluates to true.

You can set this default using the combo box available in the Choice Editor's toolbar to:

- Any of the options within the table except those you have excluded from the document.
- **THROW** to throw a Teiid Designer Server exception.
- **RECORD** to record the Teiid Designer Server exception.
- **DISCARD** to place no element within the XML instance document.

Note: You must set a default action for your choice criteria.

## 8.2.4. Recursion Editor (XML)

Some **XML schemas** define data structures that contain self-referencing elements or datatypes. When generating XML documents, such data structures can produce an endless repetition of nested tags. This self-nesting pattern is known as **recursion**.

When generating virtual documents from **XML Schema**, the Teiid Designer detects recursive data structures in the XML Schema model and halts the recursive nesting pattern after two cycles. These two cycles serve different purposes when mapping the document:

- The **first cycle** can be thought of as an "**entry condition**" for the **recursion**. The mapping class located at this node defines a normal mapping transformation like that of any other in the document model.
- The second cycle defines a mapping transformation that will be performed repeatedly until conditions are met that will halt the document instance being generated by the Teiid Designer Server. This fragment of the document model is called the recursive fragment. The mapping transformation for this fragment is no different from the first, except that you can access the

first cycle's mapping class attributes, plus you have the opportunity to specify the conditions that will halt the recursion.

You can recognize a mapping class located at the second, recursive document fragment by the looping arrow button in the top-left-hand corner of the diagram object as shown below.

When you model a virtual document based on an **XML Schema** model containing recursion, you can choose whether to treat the nested fragments as recursive. You should only use recursion when the data access pattern from your data source(s) is also recursive; in other words, when the same query transformation should be executed over and over to generate and map the nested document's data content.

By default, the Teiid Designer does not mark the recursive fragments in document models to execute recursively in the Teiid Designer Server. To take advantage of this behavior, you must open the **Recursion Editor** in the recursive mapping class *mapping transformation diagram*, mark the transformation query as recursive, and specify the recursion limit properties.

### 8.2.4.1. The Editor

The Recursion Editor lets you enable and limit recursion. The Recursion Editor button only displays on mapping classes, which have recursive patterns. For example, if you have an element named Employee which contains a element named Supervisor which itself contains an Employee element nested within it, you might need to limit the number of times the elements are nested within the document.

You can set the following conditions to limit the recursion:

- A fixed number of results to the query.
- A SQL-based criteria limit condition.
- A combination of both.

To open the Recursion Editor, click on the Recursion Editor button

### Q

on the displayed mapping class.



## Figure 8.36. Open Recursion Editor Button



## Figure 8.37. Recursion Editor

- To edit recursion properties:
  - Step 1 Click the Enable Recursion check box if you want the Teiid Designer Server to perform the query you specify to generate the nested tags within the XML document.
  - Step 2 Click the arrows beside the Count Limit box to limit the number of times to recursively perform the query. If you do not set a Limit Condition in the text area, the recursion finishes when the query reaches this limit. You can only set this limit to a maximum supported by your Teild Designer Server. For more information about this limit, contact your system administrator

- Step 3 Click the Action When Count Limit Exceeded drop down menu to instruct the Teiid Designer Server what to do if it encounters more results for the query than the count limit before it reaches the limit condition.
- Step 4 Click the Edit button to launch the SQL <u>Criteria Builder</u> to build a limiting condition for this recursion.

## Note

The Teiid Designer Server will evaluate this condition each time it recursively performs this query. If this criteria clause evaluates false, the Teiid Designer Server performs the query recursively again unless it has reached the **Count Limit**. If the criteria evaluates true, the Teiid Designer Server performs the mapping for the current level and ends its recursive loop.

When you have created the criteria, it displays in the Limit Condition box.

When the Teiid Designer Server dynamically populates your XML documents at runtime, it will use the recursion specifications you entered here.

## 8.2.5. Operation Editor

Editing of **Web Service** Operation transformations is simplified via the **Operation Editor**. When editing a Web Service model, an additional editor tab labeled "**Operation Editor**" is available. This editor, shown below is comprised of:

- **Operations** section showing a tree view of Interfaces and Operations contained within the Web Service model.
- Input Variables section providing editing of desired Input Variable declarations.
- **Procedure** section providing SQL editing of the procedure (See Mapping Web Service Operations).

Northwind_SQL_W5.xmi ×						
Operation Editor	🐼 💖 Cursor at (1, 1) 🛛 🖗 🖍 👗 💯 🍊					
Operations         Image: Northwind_SQL_Categories         Image: Northwind_SQL_CustomerCustomerDemo         Image: Northwind_SQL_CustomerCustomerDemo         Image: Northwind_SQL_CustomerDemographics         Image: Northwind_SQL_Customers         Image: Northwind_SQL_Customers         Image: Northwind_SQL_EmployeeTerritories         Image: Northwind_SQL_EmployeeTerritories         Image: Northwind_SQL_EmployeeS         Image: Northwind_SQL_Order_Details         Image: Northwind_SQL_Products         Image: Northwind_SQL_Products         Image: Northwind_SQL_Suppers         Image: Northwind_SQL_Suppers         Image: Northwind_SQL_Suppliers         Image: Northwind_SQL_Suppliers         Image: Northwind_SQL_Suppliers         Image: Northwind_SQL_Territories         Image: Northwind_SQL_Suppliers         Image: Northwind_SQL_Territories         Image: Northwind_SQL_Suppliers         Image: Northw	Input Variables         The variables representing values from the request document that are available to the selected operation's procedure.         Edit variables         IN_CUSTOMERID : string         Introduction = String         SELECT * FROM Northwind_SQL_Output_View. CustomerCustomerDemo_OutputView         WHERE         (Northwind_SQL_Output_View.CustomerCustomerDemo_OutputView.CustomerCustomerDemo_Output_Instance.CustomerID =         VARIABLES.IN_CUSTOMERTYPEID):         VARIABLES.IN_CUSTOMERTYPEID):					
🔀 Package Diagram 🥅 Table Editor 📃 Opera	ation Editor					

## Figure 8.38. Operation Editor

The **Operations** section contains all interfaces and operations currently defined in the model.

Selecting an operation will display the variables related to the input parameter's content in the **Input Variables** section and the body of its procedure (minus the CREATE VIRTUAL PROCEDURE BEGIN - END keywords and the input variable declarations and assignments) in the **Procedure section**.

When pasting in SQL, do not include the **CREATE VIRTUAL PROCEDURE BEGIN - END** keywords. Input variables will be automatically generated when the "Content via Element" property is set on an operation's input parameter. Input variables may be created, edited, or deleted using the Edit link in the **Input Variables** section, and may only represent XPath values to single attributes and elements within the input contents; other variable declarations and assignments must be typed directly into the **Procedure** section. Clicking the Edit link will display the following dialog:

🕾 Edit Input Variables		
Request Document	Input Variables	
Check a request document component to define an input variable for the Web Service procedure. Uncheck a request document component to delete an existing input variable.	Select an input variable to see the XPath that will be used to	
CORDER_DETAILS_Input : ORDER_DETAILS_Input_Type	request document.	
PRODUCTID : xs:long	IN_ORDERID : string IN_PRODUCTID : string	
XPath for IN_PRODUCTID		
/*:ORDER_DETAILS_Input/*:PRODUCTID		
	OK	

## Figure 8.39. Edit Input Variables Dialog

**Input Variables** can be deleted via right-click menu **Delete** action (see figure below) for selected variables or using **CntI-X**, **Delete** keys. You can also rename an Input Variable using the right-click menu **Rename** action or the **F2** key. (Note: Input variables names have to be named starting with the "IN\_" characters. Renaming a variable will always insure its name is prepended with "IN\_").

Inp	ut Variables	
Sele the sele req	ect an input variable to XPath that will be used of its value from the uest document.	see to
IN	_ORDERID : string	
IN	Delete	
	Rename	

Figure 8.40. Input Variable Context Menu
# **Editing Models and Projects**

#### Go To: Table of Contents [index.html]

Teiid Designer offers three basic model edit actions: **Rename**, **Move** and **Save As...** and one project-related action, **Clone Project**. These actions are described below.

## 9.1. Rename A Model

- To rename a model in your workspace:
  - Step 1 Select a model in the Section 4.1, "Model Explorer View".
  - Step 2 Right-click select the Refactor > Rename action.

E 🔁 NewEng	jland		<b>ORDERS</b>
⊕ ि 😭 🚺 WestCoast	New Child Create Web Service Open Open With Copy Copy Paste Paste Special Delete		<ul> <li>A ORDERID : long</li> <li>A CUSTOMERID : string(10)</li> <li>A EMPLOYEEID : long</li> <li>A ORDERDATE : timestamp</li> <li>A REQUIREDDATE : timestamp</li> <li>A SHIPPEDDATE : timestamp</li> <li>A SHIPVIA : long</li> <li>A FREIGHT : bigdecimal</li> <li>A SHIPNAME : string(80)</li> <li>A SHIPADDRESS : string(120)</li> <li>A SHIPCITY : string(30)</li> <li>A SHIPREGION : string(30)</li> </ul>
	Refactor Models	L	Jndo move Northwind_East.xmi ledo
	≥ Import ≧ Export	F N N	Rename Namespace URI Nove
	Refresh Compare With Replace With		
	Resource Properties		

Figure 9.1. Refactor Rename Action In Model Explorer

• Step 3 - Specify unique model name in the Rename Model File dialog. Click OK.



Figure 9.2. Rename Model File Dialog



#### Note

Renaming a model that is a dependency to another model will automatically change the model imports for those models. If source model CustomerSource is renamed to OldCustomerSource, for instance, the import statement for the view model CustomerAccounts which imports CustomerSource will be changed to reflect the new name.

## 9.2. Move Model

- To move a model in your workspace:
  - Step 1 Select a model in the Section 4.1, "Model Explorer View".
  - Step 2 Right-click select the Refactor > Move action.

### Figure 9.3. Refactor Move Action In Model Explorer

• Step 3 - Select a new location (i.e. Project or Folder) and click OK.

### Figure 9.4. Move Model Dialog

### 9.3. Save Copy of Model

The **Save As...** action performs a similar function as the **Refactor Rename** action except the renamed model a structural copy of the original model.



### Note

Each model object maintains it's own unique ID, so copying a model will result in a exact structural copy of your original model but with re-generated unique object IDs. Be aware that locating and copying your models via your local file system may result in runtime errors within Designer. Each model is expected to be unique and duplicate models are not permitted.

- To create a duplicate model using Save As...:
  - Step 1 Open the model you wish to copy in a **Model Editor** by double-clicking the model in Section 4.1, "Model Explorer View" or right-click select **Open** action.
  - Step 2 Select the editor tab for the model you opened.

	R Nort	nwind_East.xmi ×	
7	\$		
	Æ	< <base table=""/> >	

Figure 9.5. Select Editor Tab

• Step 3 - Select File > Save As... action to open the Save Model As dialog.

🗠 Save Model As	
Select a destination for the copy	
.project	
Enter the new model name	
Ready to copy	OK Cancel

### Figure 9.6. Save Model As Dialog

- Step 4 Enter a unique model name in the new model name text field and click OK.
- Step 5 If dependent models are detected, the Save Model As Import References dialog is presented to give you the opportunity to change any of the dependent models imports to reference the new model or not.

🐏 Save Model As - Import References 🛛 🛛 🔀				
The following model files use or reference the model to be copied.				
Select any that should be modified to use the new copy.				
AllCustomerAccounts/Northwind_View.xmi				
Select All Deselect All				
OK Cancel				

### Figure 9.7. Save Model As Dialog

# 9.4. Clone Project

Because each instance of of a model contains a unique ID and each object in each model contains a unique ID, copying a project is a delicate task. For this reason, the Clone Project action was created to manage the creation of exact structural copies of all models in the source project.

- The following lists specific rules and limitations for this action.
  - This action clones a complete model project containing any number of model (XMI or XSD) files organized in a user-defined directory structure.
  - All object references (UUIDs) within the original project will be replaced with new unique references.
  - Any model dependencies or internal object references are refactored to reflect the dependencies within the cloned project.
  - Any model references to models in projects external to the original project will NOT be replaced.
  - Only XMI and XSD files are cloned. All other file types in your project will NOT be processed nor copied into your newly cloned project including VDBs

- If one or more editors that require "save" are open, the user will be asked to save them before continuing with the cloning process.
- To clone a model project ::
  - Step 1 Select an existing model project in the Section 4.1, "Model Explorer View".
  - Step 2 Right-click select the Clone Project action or select the Project > Clone Project action located in the Teiid Designer's main menu bar.



Figure 9.8. Clone Project In Context Menu

	Open Project	
	Close Project	
	🗟 Build All	Ctrl+B
	Build Project	
	Build Working Set	+
	Clean	
~	Build Automatically	
	♥ Validate Model Transformations	
	Clone Project	
	<sup>4</sup> ≣ Build Project Imports	
	entric de la	
	Properties	

### Figure 9.9. Clone Project In Project Menu

• Step 3 - On the Clone Project wizard page, provide a name for your new project.

🙅 Clone Model Project	
Clone Model Project Specify cloned model project name.	
Project name: CopyOfPartsSupplier  Project contents  V Use default  Directory: D:\Designer_R050502-022208\workspace\	Browse,
Finish	Cancel

### Figure 9.10. Clone Project In Project Menu

- Step 4 (Optional) If you wish to create your cloned project in a location other than your default workspace location, uncheck the **Use default** check-box and specify (type in or browse to) a new directory location on your local file system.
- Step 5 Click Finish to generate your new project.

# **Testing Your Models**

#### Go To: Table of Contents [index.html]

As described in briefly Chapter 1, you can test your models in Teiid Designer by using the Preview Data action 5

You can also test your models via your deployable VDB. These two options will be described in detail in this chapter as well as managing your required connectors.

# **10.1. Manage Connectors**

In order to test your data, the Teiid Designer requires applicable connector information. This information is stored in the properties of a Connector. The Teiid Designer provides a <u>Connectors</u> view for managing your connectors and this section summarizes its features and capabilities.

### **10.1.1. Manage Connectors**

To edit properties of a connector, you can either select the connector and edit the properties in the **Properties** view, or select the Edit action and open the **Connector Editor** dialog presented below. Select the applicable 'Value' to activate the individual value editor.

	Connector Editor		
dit Connector			
Name is valid			
Connector Name			
Name BooksOracle Connector			
Composition Description			
Connector Properties			
		Show Advanced	d Properties
Property	Value		
JDBC URL	jdbc:oracle:thin:@myserver.a	abcbox:1521:db123	
User Name	books		
Password	**		
Class Path	classes12_g.jar		
Connection Source Class	oracle.jdbc.driver.OracleDriv	er	
Is XA	false		
L			
0			Const
(V)		OK	Cancel

### Figure 10.1. Connector Editor Dialog

### 10.1.2. Create New Connector

Create a new connector by either selecting the **New Connector** action in the <u>Connectors</u> view toolbar or right-click select the same action from the context menu. If connector types are displayed in the <u>Connectors</u> view, you can select a connector type first, then select the **New Connector** action. This will pre-select the type in the **New Connector** dialog. The action will launch the **New Connector** dialog shown below. Provide a valid connector name, select desired connector type, enter your custom properties and click **OK** to create you new connector.

	New Connector
reate New Connector	
Name is valid	
Connector Name	
Name Products	
C	
Connector Type:	
Oracle Connector	0
Connector Properties	
	C Shaw Advanced Properties
	Show Advanced Properties
Property	Value
JDBC URL	jdbc:oracle:thin:@myserver.abcbox:1521:db123
User Name	products
Password	*****
Class Path	classes12_g.jar
Connection Source Class	oracle.jdbc.driver.OracleDriver
Is XA	false
2	OK Cancel
U U	

### Figure 10.2. Create New Connector Dialog

### **10.1.3. Binding Sources to Connectors**

In order to preview your data, you'll need to bind your source models to valid connectors. If you import your source from JDBC, your connector will be created during the import process and your source automatically bound to that connector.

You can bind, unbind or change source bindings in a couple of ways.

To bind a source, you can:

- Select a model in the Section 4.1, "Model Explorer View" view select the Modeling > Connector
   > Bind action. This action presents a dialog for selecting your desired existing Connector. Select a binding and click OK.
- Select a model in the Section 4.1, "Model Explorer View" view and drag the model into the <u>Connectors</u> view and drop into the desired existing Connector.

To unbind a source:

- Select the source model in <u>Connectors</u> view and right-click select the **Unbind** action.
- Select the model in the Section 4.1, "Model Explorer View" view and select the Modeling > Connector >> Unbind action.

### **10.1.4. Importing Connectors and Types**

You can import your custom connector types and connectors into Designer. These connectors can be in the form of a **\*.cdk** file (i.e. exported from Designer) or from a **\*.caf** (connector archive file) defined during connector development.

- To import connector types and/or connectors:
  - Step 1 Select the Import Connectors (\*.cdk, \*.caf) action

located in the **Connectors** view toolbar or in the right-click menu.

Step 2 - In the first wizard page, Select the Import... button to select your \*.cdk or \*.caf file on your local file system. This page presents a summary of the custom connector types defined in your cdk/caf file. Use the check-boxes in the table to choose which types to import. Select Next> to view additional connector and jar import options or click Finish (if enabled) to import selected types and all associated connectors and connectors jars. Note: This wizard page contains a details panel which displays status messages for selected connector types.

🛢 Import Conn	ectors 🔲 🗖 🔀
Select Connec No connector typ	tor Types To Import es will be imported. Click 'Next' to select connectors to import.
Connector File:	D:\Exported Connectors\savedConnectors.cdk Import
Name	
Oracle Conne     Details	ector
Connector Typ "Oracle Conne	be "Oracle Connector" is not selected for import. (ctor" will not replace the existing workspace connector type.
(?)	< Back Next > Finish Cancel

### Figure 10.3. Select Connector Types to Import Dialog

• Step 3 - The second wizard page contains connector types defined in your cdk/caf file. Use the check-boxes in the table to choose which bindings to import. Select Next> to view additional jar import options or click **Finish** (if enabled) to import selected types, bindings and all associated connectors jars. *Note: Note: This wizard page contains a details panel which displays status messages for selected connector types.* 

🖨 Import Connectors		
Select Connectors To 0 Connector type(s) and a archive files to import.	ck next to select	
Name	Component Type	
Products Oracle	Oracle Connector	
Details Connector "Products C	racle" is selected for import.	
0	< Back Next >	Finish Cancel

### Figure 10.4. Select Connectors To Import Dialog

• Step 4 - The last wizard page contains connector jars defined either in your caf file or jars located in the same file system folder as the initial cdk file. If while importing a cdk file the required jars are not present in the selected file system folder, select the **Find...** button to locate and select the required jar from your local file system.

### Notes

i

1) This wizard page contains a details panel which displays status messages for selected connector types.

2) The **Finish** button will not enable until all custom connector jars are identified and defined. These jar files are defined in the **classPath** property for both connectors and connector types.

Import Connecto	rs		
lect Archive File connector type(s), 1 nish to import.	s To Import connector(s), and 1 archive file(s)	will be imported. Click	
Name	Import Location		
classes12_g.jar	D:\Exported Connectors\		
			e Existing
Details The required "classe	es12 .a.iar" found in the imported (	dk directory will be used	I. 🔺
			<u>×</u>
D	<back next=""></back>	Finish	Cancel

### Figure 10.5. Select Archive Files To Import Dialog

• Step 5 - Click Finish to complete the import.

### **10.1.5. Export Connectors**

You can export your custom connectors from Teiid Designer into a \*.cdk file.

- To export connectors:
  - Step 1 Select one or more Connectors in the <u>Connectors</u> view.
  - Step 2 Select the Export Connectors (\*.cdk) action

located in the Connectors view toolbar or in the right-click menu.

• Step 3 - In the wizard, select a location on your local file system and specify a \*.cdk file name (default is: savedBindings.cdk)

Save Connecto	rs To File					? 🗙
Save in:	🗀 Exported Cor	nnectors	~	O Ø D	<del>.</del>	
My Recent Documents						
Desktop						
My Documents						
My Computer						
My Network Places	File name: Save as type:	savedConnectors.c	cdk	~		ave

### Figure 10.6. Save Connectors To File Dialog

• Step 4 - Click Finish to complete the import.

# **10.2. Previewing Data For a Model**

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 <u>*Preview*</u> results view. This action is accessible throughout the Teiid Designer in various view toolbars and context menus.

There are two requirements for previewing your data: the selected object must be one of several previewable model object types and all source models within the model dependency tree must be bound to connectors. Model objects that can be previewed include: relational tables and views (including tables involving access patterns), relational procedures, Web service operations and XML document staging tables.

Note that any virtual table, view or procedure is previewable as long as all "physical" source models are bound via connectors. (See <u>Connectors</u> view )

### **Testing Your Transformations**

When transformation Transformation editing SQL in the Editor, а special Preview data action provided in the editor tool-bar is 5

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.

### **10.2.1. Preview Relational Table or View**

- To preview a relational table, relational view or staging table:
  - Step 1 Select a relational table or view in the Section 4.1, "Model Explorer View" or diagram. The table or view can be in a view model as well as a source model. Staging tables are not visible in the Section 4.1, "Model Explorer View", so you need to open the mapping diagram and select it there.
  - Step 2 Right-click select the Preview Data action

You can also select the same action in the tool-bar of either the Section 4.1, "Model Explorer View" or diagram.

• Step 3 - Your query results will be displayed in the <u>*Preview*</u> results view. The view will automatically open or get focus if not visible in your perspective.

### **10.2.2. Preview Relational Table With Access Pattern**

- To preview a relational table or view with access pattern:
  - Step 1 Select a relational table or view in the Section 4.1, "Model Explorer View" or diagram that contains an access pattern. The table or view can be in a view model as well as a source model.
  - Step 2 Right-click select the Preview Data action

You can also select the same action in the tool-bar of either the Section 4.1, "Model Explorer View" or diagram.

• Step 3 - A column input dialog is presented. Select each access pattern and enter a value for each required column. Note that if data entered does not match the column datatype (String, integer, etc...), an error message will be displayed in the dialog header. When all required values are entered, click the **OK** button to execute the query.

Fisher bata		
To preview data from this table All access pattern column values are set	<ul> <li>select an access pattern and proceeds.</li> <li>Select OK to continue.</li> </ul>	ovide a value for each required column.
Access Patterns	Required Columns	)
A NameAndID	PUBLISHER_ID : long	20022
	NAME : string(255)	Mifflin
		OK Cancel

### Figure 10.7. Access Pattern Column Input Dialog

• **Step 4** - Your query results will be displayed in the <u>*Preview*</u> results view. The view will automatically open or get focus if not visible in your perspective.

### **10.2.3. Preview Relational Procedure**

- To preview a relational procedure:
  - **Step 1** Select a relational procedure in the Section 4.1, "Model Explorer View" or diagram. The procedure can be in a view model as well as a source model.
  - Step 2 Right-click select the Preview Data action

You can also select the same action in the tool-bar of either the Section 4.1, "Model Explorer View" or diagram.

• **Step 3** - An input parameter input dialog is presented. Enter a valid value for each parameter. Note that if data entered does not match the parameter datatype (String, integer, etc...), an error message will be displayed in the dialog header. When all required values are entered, click the **OK** button to execute the query.

🙅 Preview Data				
To execute this All parameter valu	s <b>procedure, inpu</b> es are set. Select OK t	i <b>t values are r</b> o continue.	equired.	1
-Input Parame	ters blue			
			ОК	Cancel

### Figure 10.8. Procedure Parameter Input Dialog

• **Step 4** - Your query results will be displayed in the <u>*Preview*</u> results view. The view will automatically open or get focus if not visible in your perspective.

### 10.2.4. Preview Web Service Operation

- To preview a Web service operation:
  - **Step 1** Select a Web service operation in the *Section 4.1, "Model Explorer View*" or diagram. The operation can be in a view model as well as a source model.
  - Step 2 Right-click select the Preview Data action

You can also select the same action in the tool-bar of either the Section 4.1, "Model Explorer View" or diagram.

• **Step 3** - An input parameter input dialog is presented. Enter a valid value for each parameter. Note that if data entered does not match the parameter datatype (String, integer, etc...), an error message will be displayed in the dialog header. When all required values are entered, click the **OK** button to execute the query.

🙅 Preview Data	
To execute this proc All parameter values are s	edure, input values are required. et. Select OK to continue.
- Input Parameters - AUTHOR_ID : xs:long	10022
	OK Cancel

### Figure 10.9. Procedure Parameter Input Dialog

• **Step 4** - Your query results will be displayed in the <u>*Preview*</u> results view. The view will automatically open or get focus if not visible in your perspective.

## **10.3. Testing With Your VDB**

- To execute a query against your VDB:
  - Step 1 In the Section 4.1, "Model Explorer View", double-click a VDB, or select one and choose Open action to open a <u>VDB Editor</u>.
  - Step 2 Select the Execute tab. If connectors are missing for any source models, the Execute button will be disabled. Edit your connectors to create/select connectors for your models.

Execution Status Execute The virtual database	se is ready to execute.	
xecution Options ansaction AutoWrap: OPTIM:		
onnectors elect row below to view or edit s	specific connector properties	
Source Model	Connector	Edit
Products_SQL_Server	Products_SQL_Server Connector - SQL Server Connector	Unbind
Connector Properties:		Propercies
Fruberty	value	
IDBC LIDI	jabelsqiser ver (//sincabzo.dac.in/company.com.1455).dacabaseriame—produ	
JDBC URL	Droducte	
JDBC URL User Name Password	**	
JDBC URL User Name Password Class Path	extensioniar:solidbc4.iar	
JDBC URL User Name Password Class Path Connection Source Class	produccs *** extensionjar:sqljdbc4.jar com.microsoft.sglserver.jdbc.SQLServerDriver	
JDBC URL User Name Password Class Path Connection Source Class Is XA	produccs ** extensionjar:sqljdbc4.jar com.microsoft.sqlserver.jdbc.SQLServerDriver false	

### Figure 10.10. Connections View

• Step 3 - Select the Execute button to establish a connection and open the <u>SQL Explorer</u> perspective. Your VDB should now be connected via the connectors defined previously. Verify the connection in the Connections view.



### Figure 10.11. Connections View

• Step 4 - In the Database Structure view (upper left), expand the Table node to list the tables in your VDB.

artsSuppli E E Dat E E E	er_VDB.vd abase PartsSupp	b-13:34:49	Aug 03			
日 造 Dat 日 造 日 王	abase PartsSupp	diar UDD				
□       □						
Columns	Indexes	Primary Ke	ey   Forei	gn Key	Preview	Row Count
Colu	Data	Size	Deci	Defa	. Acce	. Com
SHIPP	short	2	0		YES	
SHIPP	string	30	0		NO	

### Figure 10.12. Database Structure View

• Step 5 - Select one of the tables and do a right click. Select the Generate Select in SQL Editor menu item.

artsSupplier_ = E Databa = E Pa = E E E = E	VDB. vdl pase artsSupp Procece System Table Pa Pa Pa Pa Pa Pa Pa Pa Pa Pa	b-13:34:49 dure artsVirtual.p m artsSupplier artsSupplier artsSupplier artSupplier artSupplier artSupplier	P Aug 03 partsByCo r_SourceA r_SourceA r_SourceA r_SourceA _SourceB. _SourceB. _SourceB	a.PARTS A.SHIP_VI A.SUPPLIE A.SUPPLIE PARTS SUID_UTA Select in	IA ER ER_PARTS SQL Editor	
E E Databa E E Pa E E E E	artsSupp Procec E Procec Syster Table Pa Pa Pa Pa Pa Pa Pa Pa Pa Pa	olier_VDB dure artsVirtual.p m artsSupplier artsSupplier artsSupplier artSupplier artSupplier artSupplier	partsByCo r_SourceA r_SourceA r_SourceA r_SourceA r_SourceA SourceB. SourceB.	A.PARTS A.SHIP_VJ A.STATUS A.SUPPLIE PARTS SUID_UTA Select in	IA ER ER_PARTS SQL Editor	
	<ul> <li>➡ Database</li> <li>➡ PartsSupplier_VDB</li> <li>➡ Procedure</li> <li>➡ PartsVirtual.partsByColor</li> <li>➡ System</li> <li>➡ Table</li> <li>PartsSupplier_SourceA.PARTS</li> <li>PartsSupplier_SourceA.SHIP_VIA</li> <li>PartsSupplier_SourceA.STATUS</li> <li>PartsSupplier_SourceA.SUPPLIER</li> <li>PartSupplier_SourceA.SUPPLIER_PARTS</li> <li>PartSupplier_SourceB.PARTS</li> </ul>					
Columns In	ndexes	Primary Ke	ey   Forei	ign Key 📔	Preview   I	Row Count
Colu Da	ata	Size	Deci	Defa	Acce	Com
SHIPP sh	hort	2	0		YES	
SHIPP st	tring	30	0		NO	

### Figure 10.13. Generate SELECT SQL Action

• Step 6 - A SELECT statement will appear in the SQL Editor (upper right).



#### Figure 10.14. Generated SQL

• Step 7 - Click the Execute SQL icon

to run the query. The results will appear in the SQL Results tab (lower right).

	ce_VDB.vdb 🛛 🚺 BooksWebService_VDB.vdb	
lect ISBN, BooksWebSe	TITLE, SUBTITLE, PUBLISHER, PUBLISH_YEAR, EDI rvice_VDB"."Books_SourceB.BOOKS"	TION, TYPE from
SQL Results 🗙	41 of 41 rows	📑 🖩 🖟 🍫 😫 🗎 🚰 <sup></sup>
ISBN	TITLE	SUBTITLE
0-7356-0877-6		
	After the Gold Rush	Creating a True Profession of Soft
0-201-43287-0	After the Gold Rush Automated Software Testing	Creating a True Profession of Soft Introduction, Management, and P
0-201-43287-0	After the Gold Rush Automated Software Testing Code Complete	Creating a True Profession of Sofi Introduction, Management, and P A Practical Handbook of Software
0-201-43287-0 1-55615-484-4 0-201-10088-6	After the Gold Rush Automated Software Testing Code Complete Compilers	Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools
D-201-43287-0 1-55615-484-4 D-201-10088-6 D-7897-2271-2	After the Gold Rush Automated Software Testing Code Compilete Compilers Complete Idiot's Guide to Project Management with Microsoft Project 2000	Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools
0-201-43287-0 1-55615-484-4 0-201-10088-6 0-7897-2271-2 0-471-98710-7	After the Gold Rush Automated Software Testing Code Complete Compilers Complete Idiot's Guide to Project Management with Microsoft Project 2000 Concurrency State Models & Java Programs	Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software – Principles, Techniques, and Tools
0-201-43287-0 1-55615-484-4 0-201-10088-6 0-7897-2271-2 0-471-98710-7 0-201-31009-0	After the Gold Rush Automated Software Testing Code Complete Compilers Complete Idiot's Guide to Project Management with Microsoft Project 2000 Concurrency State Models & Java Programs Concurrent Programming in Java	Creating a True Profession of Sofi Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns
D-201-43287-0 1-55615-484-4 D-201-10088-6 D-7897-2271-2 D-471-98710-7 D-201-31009-0 D-932633-33-1	After the Gold Rush Automated Software Testing Code Complete Compilers Complete Idiot's Guide to Project Management with Microsoft Project 2000 Concurrency State Models & Java Programs Concurrent Programming in Java Creating a Software Engineering Culture	Creating a True Profession of Sofi Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns
D-201-43287-0 1-55615-484-4 D-201-10088-6 D-7897-2271-2 D-471-98710-7 D-201-31009-0 D-932633-33-1 1558605150	After the Gold Rush Automated Software Testing Code Complete Compilers Complete Idiot's Guide to Project Management with Microsoft Project 2000 Concurrency State Models & Java Programs Concurrent Programming in Java Creating a Software Engineering Culture Database Design for Smarties	Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns Using UML for Data Modeling
D-201-43287-0 1-55615-484-4 D-201-10088-6 D-7897-2271-2 D-471-98710-7 0-201-31009-0 D-932633-33-1 1558605150 D-13-239856-7	After the Gold Rush Automated Software Testing Code Compilete Compilers Complete Idiot's Guide to Project Management with Microsoft Project 2000 Concurrency State Models & Java Programs Concurrent Programming in Java Creating a Software Engineering Culture Database Design for Smarties How to Run Successful Projects II	Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns Using UML for Data Modeling
0-201-43287-0 1-55615-484-4 0-201-10088-6 0-7897-2271-2 0-471-98710-7 0-201-31009-0 0-932633-33-1 1558605150 0-13-239856-7 0-201-31000-7	After the Gold Rush Automated Software Testing Code Compilets Compilets Compilete Idiot's Guide to Project Management with Microsoft Project 2000 Concurrency State Models & Java Programs Concurrent Programming in Java Creating a Software Engineering Culture Database Design for Smarties How to Run Successful Projects II Inside Java 2 Security	Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns Using UML for Data Modeling
0-201-43287-0 1-55615-484-4 0-201-10088-6 0-7897-2271-2 0-471-98710-7 0-201-31009-0 0-932633-33-1 1558605150 0-13-239856-7 0-201-31000-7 0-596-00016-2	After the Gold Rush Automated Software Testing Code Complete Compilers Complete Idiot's Guide to Project Management with Microsoft Project 2000 Concurrency State Models & Java Programs Concurrent Programming in Java Creating a Software Engineering Culture Database Design for Smarties How to Run Successful Projects II Inside Java 2 Security Java and YMI	Creating a True Profession of Soft Introduction, Management, and P A Practical Handbook of Software Principles, Techniques, and Tools Design Principles and Patterns Using UML for Data Modeling

### Figure 10.15. SQL Results View

When executing queries on XML View Documents, the results are displayed in XML format as shown below:

K 🗁 🖉_		
elect * from "BooksWebService	_VDB"."BooksXML.bookCollectionDocument"	
QL Results 🔄 XML Results 🗙	1 of 1 documents 🛛 📋 🎼 🏇 🔰	K 🗈 🕋 🗌
2		
2		
<pre><rr></rr></pre>		~
<pre><?xml version="1.0" encoding="UTF-8"?> <books:bookcollection xmlns:books="http://www&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;v.metamatrix.com/XMLSchema/DataSets/Books"></books:bookcollection></pre>	^	
xml version="1.0" encoding="UTF-8"? <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www&lt;br&gt;&lt;book isbn="></books:bookcollection>	v.metamatrix.com/XMLSchema/DataSets/Books">	
xml version="1.0" encoding="UTF-8"? <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www&lt;br&gt;&lt;book isbn="> <title>After the Gold Rush</title></books:bookcollection>	v.metamatrix.com/XMLSchema/DataSets/Books">	
<pre><?xml version="1.0" encoding="UTF-8"?> <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www &lt;book isbn="> <title>After the Gold Rush</title> <subtitle>Creating a True Profession of Softw</subtitle></books:bookcollection></pre>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	
<pre><?xml version="1.0" encoding="UTF-8"?> &lt;8ooks:bookCollection xmlns:Books="http://www <book isbn="0-7356-0877-6"> <title>After the Gold Rush </title> <subtitle>Creating a True Profession of Softw <edition>1</edition></subtitle></book></pre>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	
xml version="1.0" encoding="UTF-8"? <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www&lt;br&gt;&lt;book isbn="> <title>After the Gold Rush</title> <subtitle>Creating a True Profession of Softw <edition>1</edition> <authors></authors></subtitle></books:bookcollection>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	
xml version="1.0" encoding="UTF-8"? <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www&lt;br&gt;&lt;book isbn="> <title>After the Gold Rush</title> <subtitle>Creating a True Profession of Softw <edition>1 </edition> <authors> <author>McConnell</author></authors></subtitle></books:bookcollection>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	
<pre><?xml version="1.0" encoding="UTF-8"?> <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www &lt;book isbn="> <title>After the Gold Rush</title> <subtile>Creating a True Profession of Softw <edition>1</edition> <authors> <authors> </authors></authors><td>v.metamatrix.com/XMLSchema/DataSets/Books"&gt; ware Engineering</td><td></td></subtile></books:bookcollection></pre>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	
<pre><?xml version="1.0" encoding="UTF-8"?> <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www &lt;book isbn="> <title>After the Gold Rush</title> <subtitle>Creating a True Profession of Softw <edition>1</edition> <authors> <authors> <authors> <publishinginformation></publishinginformation></authors></authors></authors></subtitle></books:bookcollection></pre>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	
<pre><?xml version="1.0" encoding="UTF-8"?> <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www &lt;book isbn="> <title>After the Gold Rush &lt;{title&gt; <subtitle>Creating a True Profession of Softw <edition>1</edition> <authors> <authors> </authors>  <publisher>Microsoft Press</publisher></authors></subtitle></title></books:bookcollection></pre>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	
xml version="1.0" encoding="UTF-8"? <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www&lt;br&gt;&lt;book isbn="> <title>After the Gold Rush</title> <subtitle>Creating a True Profession of Softw <edition>1</edition> <authors> <authors> </authors>  <publisher>Microsoft Press</publisher> <publishdate>1999</publishdate></authors></subtitle></books:bookcollection>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	
<pre><?xml version="1.0" encoding="UTF-8"?> <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www &lt;book isbn="> <title>After the Gold Rush</title> <subtitle>Creating a True Profession of Softw <edition>1</edition> <authors> <authors> <authors> <publishinginformation> <publisher>Microsoft Press</publisher> <publishdate>1999</publishdate> </publishinginformation></authors></authors></authors></subtitle></books:bookcollection></pre>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	
<pre><?xml version="1.0" encoding="UTF-8"?> <books:bookcollection 0-7356-0877-6"="" xmlns:books="http://www &lt;book isbn="> <title>After the Gold Rush</title> <subtitle>Creating a True Profession of Softw <edition>1</edition> <authors> <autho< td=""><td>v.metamatrix.com/XMLSchema/DataSets/Books"&gt; ware Engineering</td></autho<></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></authors></subtitle></books:bookcollection></pre>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	
<pre><?xml version="1.0" encoding="UTF-8"?> &lt;8ooks:bookCollection xmlns:Books="http://www <book isbn="0-7356-0877-6"> <title>After the Gold Rush</title> <subtitle>Creating a True Profession of Softw <edition>1</edition> <authors> <authors>McConnell </authors> <publishinginformation> <publisher>Microsoft Press</publisher> <publishdate>1999</publishdate> </publishinginformation>   <td>v.metamatrix.com/XMLSchema/DataSets/Books"&gt; ware Engineering</td></authors></subtitle></book></pre>	v.metamatrix.com/XMLSchema/DataSets/Books"> ware Engineering	

Figure 10.16. XML SQL Results View

When each query is executed, the SQL is stored in the **SQL History** table so the queries can be re-submitted without re-typing it.

	5QL History 🖾	
ID	Query	
1	select PART_ID, PART_NAME, PART_COL	.OR, PART_WEIGH1
2	select SHIPPER_ID, SHIPPER_NAME from	"PartsSupplier_VDE
<	III	>

### Figure 10.17. SQL History View

This content of this **SQL History** table can be edited via right-click menu shown below.

	SQL History 🔀		<b>≩</b> □ □
ID	Query		
1	select ISBN, TITLE, SUB	TITLE PUBLISHER PUBLIS	H_YEAR, EI
2	select * from "Booksy	Open in SQL Editor	bookCollec
		Remove from history	
		Clear history records	
		Copy to Clipboard	
<	IIII I		>

### Figure 10.18. SQL History View Action Menu

### **Saving Your Query Results**

Query results, both SQL and XML, can be saved to text files in the SQL Explorer perspective.

The toolbars for **SQL Results** and **XML Results** views contains an additional action/button (see below) which presents a file save dialog. Note that the action can also be accessed via the right-click menu on both views.

🗟 SQL Results 🗙	16 of 16 rows	🔲 🖩 🖟 🔅 🗱 🚺
1 10 1		

### Figure 10.19. Save SQL Results Action

In addition, this file save dialog is presented to the user when the XML Document results exceed the character size limit defined by a user preference property for the SQL Explorer.

SQLExplorer	\$ • \$ •
Preview - Limit Rows	80
SQL - Limit Rows	2000
AutoCommit	
🗖 Commit On Close	
Tables and columns auto	-completing assistance. Use only with fast database connections
Generate query plans (re	equires connection restart)
XML Results Display limit (# c	:hars) 100000

### Figure 10.20. XML Document Query Character Limit

When this limit is exceeded, the user is presented with a dialog to either save the results to a file or continue displaying the results and possibly incurring substantial runtime memory allocation.

🥸 Larg	ge XML Document Results Detected				
?	The XML document results from you query contain over 10000 characters. (See Window > Preferences > SQLExplorer to change limit)				
	It is recommended that you save your results and view with external editor.				
	Do you wish to save results to file?				
	Yes No Cancel				

### Figure 10.21. Large XML Document Detected Dialog

When the Save option is accepted, the XML Query Results view displays the file name and path of the exported file.



Figure 10.22. Large XML Document Saved Results Message

# Searching

#### Go To: Table of Contents [index.html]

The Teiid Designer provides multiple search actions located in the main toolbar's Search menu.



### Figure 11.1. Search Options

• The individual actions in the **Search** menu are described below:

• 🖋

**Search...** - Launches the standard Search dialog, which can be used to find resource containing specific text or expressions and filter based on file name patterns. For standard Eclipse installations, the default search tile will be File Search. (See Eclipse documentation for details).

• 🔗

**File...** - Launches the standard Search dialog, which can be used to find resource containing specific text or expressions and filter based on file name patterns. The File Search tab will be selected in the dialog. (See Eclipse documentation for details).

• 🔎

**Find Model Object** - Launches the **Find Model Object** dialog, which can be used to locate an object in the workspace by specifying all or part of its name. Selecting the object will open it in the appropriate editor.

• 🛃

**Relationship...** - Launches the Search dialog and auto-selects the Relationships tab. Users can search for models in the workspace by specifying a relationship type, participant locations, and/or names containing specified text. Search results appear in the <u>Search</u> <u>Results</u> view, and double-clicking a result will open that model in the appropriate editor.

• 🚀

**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.

A

**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 <u>Search Results</u> view, and double-clicking a result will open that model in the appropriate editor.

# **11.1. Finding Model Objects**

The Teiid Designer provides a name-based search capability to quickly locate and display model objects.

- To find a model object:
  - Step 1 Open the Find Model Object dialog by either selecting the

action on the main Teiid Designer tool-bar.



### Figure 11.2. Find Model Object Action In Toolbar

or select the same action via the main menu's Search > Find Model Object

action.

🙅 Find Model Object	
Type Object Name (? = any character, * = any String);	
Books	
Matching Model Objects:	
E BOOKS S Books J Books - Oracle 9i Books - SQL Server 2000	
<ul> <li>A - Books_Oracle</li> <li>4 - Books_SQL</li> <li>4 = Books_Doc</li> </ul>	<b>&gt;</b>
Locations:	
//Books/Books_Oracle.xmi/BOOK5     //Books/Books_Oracle.xmi/BOOK5/BOOK5     //Books/Books_SQL.xmi/BOOK5	
	>
ОК	Cancel

### Figure 11.3. Find Model Object Dialog

- Step 2 Begin typing a word or partial word in the Type Object Name field. Wild-card (\*) characters will be honored. As you type, the objects which match the desired name will be displayed in the Matching Model Objects list. If there are more than one objects with the same name, the locations or paths of the objects are displayed in the Locations list.
- Step 3 If more than one object exists with the desired name, select the one of the locations.
- Step 4 Click OK. If editor is not open for the object's model, an editor will open. The desired object should end up displayed in a diagram (if applicable) and selected.

# **11.2. Search Models Via Relationship Properties**

The Teiid Designer provides a search capability to find model objects that are characterized by one or more relationship property values including relationship type, participants and name.

• To search for relationships:

• Step 1 - Select Search > Relationships... action on the Teiid Designer main menu

which opens the Search dialog.

Search 🛛 🔀
💖 File Search 📝 Metadata 💀 Relationship
Туре
Any
C Name:
₩ Include subtypes
Participants Found In
Any Resource
C Selected Resources (0 selected participant resources)
(* = any string, ? = any character, = escape for literals: * ? )
Workspace C Selected resources
C Working set: Choose
Customize Search Cancel

#### Figure 11.4. Relationship Search Dialog

- Step 2 Specify desired search options for Object Type, location of Participants and Name.
- Step 3 Click Search. The search will be performed and the results will be displayed in the Search Results View. If the view is not yet open, it will be opened automatically.

# **11.3. Search Transformation SQL**

The Teiid Designer provides a search capability to string values present in transformation SQL text.

- To search for string values in your transformations SQL:
  - Step 1 Select Search > Transformations... action on the Teiid Designer main menu

which opens the Search Transformations dialog.

💀 Search Tranformations 🛛 🛛 🔀
Input Find: Parts
Perform Search
Search Status Search complete for: [Parts]
Virtual Tables, Views, Procedures Operations [15] SQL matches for: [Parts]
Image: Parts < Table > : /PartsProjectA/Views/Parts_V.xmi         Image: Parts Structure         Image: Parts Parts Project A/Views/Parts_V.xmi         Image: Parts Parts Parts Project A/Views/Parts_V.xmi
Image: Supplier PARTS <table> : /PartsProjectA/Views/Parts_V.xmi         Image: Supplier PARTS <procedure> : /PartsProjectA/Views/Parts_O9I_WS.xmi/Parts_O9I_PARTS         Image: Supplier PARTS          Image: PARTS</procedure></table>
SELECT SQL String
CREATE VIRTUAL PROCEDURE BEGIN DECLARE string VARIABLES.IN_PART_ID; VARIABLES.IN_PART_ID = xPathValue(Parts_O9i_WS.Parts_O9i_PARTS.getPARTS.PARTS_InputMsg, '/*:PARTS_Input/*:PART_ID'); SELECT * FROM Parts_O9i_Output_View.PARTS_OutputView WHERE Parts_O9i_Output_View.PARTS_OutputView.PARTS_Output_Instance.PART_ID = VARIABLES.IN_PART_ID;
Edit Close

### Figure 11.5. Search Transformations Dialog

- Step 2 Specify a string segment in the Find: field and specify/change your case sensitive preference.
- Step 3 Select Perform Search button. Any transformation object containing SQL text which contains occurances of your string will be displayed in the results section.

You can select individual objects and view the SQL. If a table or view supports updates and there is insert, update or delete SQL present, you can expand the object and select the individual SQL type as shown below.

🙅 Search Tranformations	
Input       Find:     Parts       Image: Case Sensitive	•
Perform Search Search Status Search complete for: [Parts]	
Virtual Tables, Views, Procedures Operations [15] SQL matches for: [Parts]	
SUPPLIER <table> : /PartsProjectA/Views/Parts_V.xmi     SUPPLIER_PARTS <table> : /PartsProjectA/Views/Parts_V.xmi     SELECT     UPDATE     UPDATE</table></table>	
Occure     Occure	>
INSERT SQL String CREATE PROCEDURE BEGIN ROWS_UPDATED = INSERT INTO Parts_O9i.SUPPLIER_PARTS (Parts_O9i.SUPPLIER_PARTS.SUPPLIER_ID, Parts_O9i.SUPPLIER_PARTS.PART_ID, Parts_O9i.SUPPLIER_PARTS.QUANTITY, Parts_O9i.SUPPLIER_PARTS.SHIPPER_ID, Parts_O9i.SUPPLIER_PARTS.AVAILABLE_QTY, Parts_O9i.SUPPLIER_PARTS.RESERVED_QTY) VALUES (INPUT.SUPPLIER_ID, INPUT.PART_ID, INPUT.QUANTITY, INPUT.SHIPPER_ID, INPUT.AVAILABLE_QTY, INPUT.RESERVED_QTY); END	×
Edit C	lose

### Figure 11.6. Insert SQL Example

If you wish to view the selected object and its SQL in a **Model Editor**, you can click the **Edit** button. An editor will be opened if not already open. If an editor is open its tab will be selected. In addition, the **Transformation Editor** will be opened and you can perform **Find/Replace** (Ctrl-F) actions to highlight your original searched text string and edit your SQL if you wish.

# **11.4. Search Models Via Metadata Properties**

The Teiid Designer provides a search capability to find model objects that are characterized by one or more metadata property values.

- To search your models using metadata:
  - Step 1 Select Search > Metadata... action on the main Teiid Designer toolbar

which opens the **Search** dialog.

Search	
Pile Search	🔗 Metadata 🛛 🐶 Relationship 🛛
-Obiect Type	1.,
Any	
C Object Class	:
Data Type	
Any	
C Simple Type:	
☐ Include :	subtypes
C Runtime Typ	e: bigdecimal
Properties	
₩ Where:	Name
Contains Search Tex	: C Does NOT Contain F Exact Match t (case insensitive, * = any string, ? = any character):
Scope	C Selected recources
working set:	Choose
ustomize	Search Cancel

### Figure 11.7. Metadata Search Dialog

- Step 2 Specify desired search options for Object Type, Data Type and Properties.
- Step 3 Click Search. The search will be performed and the results will be displayed in the Search Results View. If the view is not yet open, it will be opened automatically.

# **Relationships**

#### Go To: Table of Contents [index.html]

### Introduction

The Teiid Designer includes the ability to create named relationships between almost any two or more modeling elements.

The **Relationship** metamodel is one way to define arbitrary associations between two or more metadata entities. Using the Relationship metamodel, the Teiid Designer provides users the ability to view, add, and edit associations between metadata entities.

But the **Generic Relationships** functionality can define relationships between metadata entities from different metamodels. For example, it is possible to create a generalized relationship between a **Relational Column** and a **UML Package**.

Users can even define their own *relationship types* that constrain the numbers and types (metaclasses) of objects that can be related.

Once the existing relationships are defined, the Teiid Designer provides a number of mechanisms that will make use of those relationships. This includes navigating to a related object, searching for relationships based on some criteria, or diagramming the relationship of various sets of objects

# **12.1. Creating Relationships**

- To create a relationship in a diagram:
  - Step 1 First select the source element of the relationship.
  - Step 2 Then Shift+Click on the second target element. Right-click and select New Association > Relationship.

R North	wind_Oracle_2.xmi × 🕞 Customer	Relationships.xmi
	< <schema>&gt; TempSchema</schema>	< <base table=""/> > SHIPPERS
		PK_SHIPPERS
	A CUSTOMERI New Child A EMPLOYEEIE New Sibling	
	ORDERDATE New Association     A REQUIREDD     A SHIPPEDDA1      Undo Relationsh	ip *1. Relationship
	A SHIPVIA : loi 🗠 Redo	ODDEDTD + land

Figure 12.1. Create New Relationship Action in Diagram

• Step 3 - In the Create Relationship dialog, supply a valid name for your relationship, Browse... and select a relationship type.

🐏 Create	Relationshi	р				<
Create a I Press Finish	n <b>ew Relatic</b> In to create the	<b>onship</b> Relationship.				
Name: Type: Location: Participar	OrderToShip Dependency /AllCustomer	per Accounts/CustomerRelatio	nships.xmi		Browse Browse	
client				supplier		
Name	ERID : long	Path   AllCustomerAccou	> <	Name SHIPPERID : long	Path /AllCustomerAccounts/	
	Add	Remove	<u></u>	Add	Remove	
					Finish Cancel	

Figure 12.2. New Relationships Wizard

• Step 4 - Click the Browse... button adjacent to the Type field and open the Select Relationship Type dialog and select an existing relationship type that best describes your relationship.

🙅 Select Relationship Type	
Select a relationship type	
Built In Types     Generic     Any     Causality     Composition     T Consumption     Consumption     Consumption     Consumption     Consumption     Compliance     CMLExtensionsProject	
Properties OK	Cancel

### Figure 12.3. Select Relationship Type Dialog

 Step 5 - Click the Browse... button adjacent to the Location field and open the Relationship Model Selector Dialog. If a relationship model already exists, you can select it and click OK.
 If not, select a valid project or folder and provide a valid name for a new relationship model which will be created for you, then click OK.

🙅 Relationship Model Selector Dialog				
	>			
.project CustomerAccounts.vdb				
Northwind_Oracle.xmi     Northwind_Oracle_2.xmi				
.project 				
WestCoastSales     .project				
AMLFileConnectorExtensions.xmi      AMLHthrConnectorExtensions.xmi	~			
New Model Name: CustomerRelationships				
Click OK to create the relationship model				
Properties OK	Cancel			

### Figure 12.4. Select Relationship Model Dialog

 Step 6 - If you wish to add additional objects to either your source or target participants, click the Add... button to display the Model Object Finder dialog. You can either browse your workspace (Workspace tab) and select objects within models, or click on the Finder tab and type in known object names to search for specific objects by name. Click OK to add selected objects to the participant lists.


Figure 12.5. Model Object Finder Dialog - Workspace Tab

🎂 Model Object Finder Dialog 📃 🗖 🔯
Workspace Finder
Type Object Name (? = any character, * = any String);
Customer
Matching Model Objects:
CUSTOMERDEMOGRAPHICS
CUSTOMERDESC 📃
CUSTOMERID
Locations:
🛛 /AllCustomerAccounts/Northwind_Oracle.xmi/CUSTOMERCUSTOMERDEMO/CUS
AllCustomerAccounts/Northwind_Oracle.xmi/CUSTOMERS/CUSTOMERID
🖄 /AllCustomerAccounts/Northwind_Oracle.xmi/ORDERS/CUSTOMERID
AllCustomerAccounts/Northwind_Oracle_2.xmi/CUSTOMERCUSTOMERDEMO/CL
AllCustomerAccounts/Northwind_Oracle_2.xmi/CUSTOMERS/CUSTOMERID
AllCustomerAccounts/Northwind_Oracle_2.xmi/ORDER5/CUSTOMERID
AllCustomerAccounts/Northwind_View.xmi/CUSTOMERCUSTOMERDEMO/CUSTC
A /AICustomerAccounts/Merthuind, View smil/CLISTOMEDS/CLISTOMEDID
Properties OK Cancel

### Figure 12.6. Model Object Finder Dialog - Finder Tab

• Step 7 - You can also edit your relationship properties by selecting the **Properties** tab in the **Create Relationship** dialog.

🙅 Model Object Finder Dialog 📃 🗖 🔯
Workspace Finder
Type Object Name (? = any character, * = any String);
Customer
Matching Model Objects:
CUSTOMERDESC 📃
AllCustomerAccounts/Northwind_Oracle.xmi/CUSTOMERCUSTOMERDEMO/CUS
AllCustomerAccounts/Northwind_Oracle.xmi/CUSTOMERS/CUSTOMERID
AllCustomerAccounts/Northwind_Oracle.xm/ORDERS/CUSTOMERID
AllCustomerAccounts/Northwind_Oracle_2.xmi/CUSTOMERCOSTOMERCERD/CL
AllCustomerAccounts/Northwind_Oracle_2.xmi/ORDERS/CUSTOMERID
AllCustomerAccounts/Northwind View.xmi/CUSTOMERCUSTOMERDEMO/CUSTC
AllCustomerAssounts/Northwind View wmitCUSTOMEDS/CUSTOMEDID
Properties OK Cancel

# Figure 12.7. Create Relationship Dialog - Properties Tab

• **Step 8** - Click **Finish** to complete the relationship. The new relationship will be displayed in a relationship diagram.



#### Figure 12.8. New Relationship In Diagram

To create a relationship from the Section 4.1, "Model Explorer View" tree, select two or more objects in the tree, right-click select **New Association > Relationship** action and follow **Steps 3 thru 8** described above.

🔁 Model Explorer 🗙 Outline 🗖 🗖 🖓 Northwind,	_Oracle_2.xmi 🛛 🧊 CustomerR
Image: Second secon	<schema>&gt; TempSchema</schema>
ORDERID : long     ORDERID : long     A CUSTC New Child     A EMPLC New Sibling     A ORDEF New Association     A REQUI	CRDERS      ORDERS      O
Image: Shippe       ✓ Undo Relationship         Image: Shippe       ✓ Redo         Image: Shippe       ✓ Cut         Image: Shippe       ✓ Copy         Image: Shippe       ✓ Copy         Image: Shippe       ✓ Cope	REQUIREDDATE : timestamp SHIPPEDDATE : timestamp SHIPPEDDATE : timestamp SHIPVIA : long FREIGHT : bigdecimal SHIPNAME : string(80) SHIPADDRESS : string(120) SHIPCITY : string(30)
SHIPC     Copy Name     FK_OR     FK_OR     FF_FK_OR     FF_FK_OR     FF_FK_OR     P FK_OR     P FK_OR     P ORDER_DE     Open     Ctrl+0	SHIPREGION : string(30)
PRODUCTS     Edit     Edit     Edit     Set Datatype     Set Datatype     SHIPPERS     SHIPPERS     SHIPPERS     Open in Relationship Navigator     OPP     PHONE   Add To New Custom Diagram	PK_ORDERS

### Figure 12.9. New Relationship Action in Model Explorer

# 12.2. Creating your own Relationship Type

- To create a new relationship type:
  - Step 1 Select an existing Relationship model in the Section 4.1, "Model Explorer View" or a folder within the model. If a Relationship model does not yet exist, create one. See <u>New</u> <u>Model Wizard</u>.
  - Step 2 Right-click select the New Child > Relationship Type action.

🚰 Model Explorer 🗙 Outline	[] <mark>ª</mark> 🤣 ⇔ ⇔	ଢ		FamilyRelatior
AllCustomerAccounts	1		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
⊕ f import declaration	New	+		
- 🦓 Relationship Diagr	New Child	•	🖳 File Reference	
	Open Open With	•	💼 Folder 😘 Relationship	
- Role B: Son			🍸 Relationship Type	
⊕ থা Married_Couple ⊕ থা Mother	Paste		:// URI Reference	
	Paste Special Delete		🐮 Custom Diagram 🔀 Custom Relationship Diag	gram
Jones_And_Daugi	Rename			

### Figure 12.10. New Relationship Type Action in Model Explorer

• Step 3 - In the Create Relationship Type dialog, supply a name for your new relationship type.

te a new R	elationship Typ	e			
s Finish to crea	te the Relationship `	Туре.			
Name:	Son				
Points of View					
The labels	for the relationship	type from the point o	of view of each role:		
Son	child of				Parent
Parent	parent to				Son
Role B Nar	me:   Parent	ered 🔽 Unique	Lower Bound: 1	Properties	-1
Role B Nar	ilasses	ered 🔽 Unique	Lower Bound:     1       Image: Navigable     Role       Exclude Metacle	e Properties	-1
Role B Nar nclude Metad	classes	ered 🔽 Unique Model Class	Lower Bound: 1  Vavigable Role  Exclude Metacla  Metaclass	asses Upper Bound: Model Class	-1
Role B Nar nclude Metad Metaclass	n.metamatrix.me	ered Vunique Model Class People	Lower Bound: 1  Navigable Role  Exclude Metacla  Metaclass	asses Model Class	-1
Role B Nar nclude Metad Metaclass Person <cor< td=""><td>n.metamatrix.me</td><td>ered Vunique Model Class People</td><td>Lower Bound: 1</td><td>Properties</td><td>-1</td></cor<>	n.metamatrix.me	ered Vunique Model Class People	Lower Bound: 1	Properties	-1
Role B Nar nclude Metad Metaclass	n.metamatrix.me	ered Vinique Model Class People	Lower Bound:       1         Image: Navigable       Role         Exclude Metaclass       Metaclass	asses Model Class	-1
Role B Nar nclude Metad Metaclass Person <cor< td=""><td>In.metamatrix.me</td><td>ered Vinique Model Class People</td><td>Lower Bound:       1         ✓       Navigable       Role         Exclude Metaclass       Metaclass</td><td>Add Remove</td><td>-1</td></cor<>	In.metamatrix.me	ered Vinique Model Class People	Lower Bound:       1         ✓       Navigable       Role         Exclude Metaclass       Metaclass	Add Remove	-1
Role B Nar	IIII  Add Rem	ered Vilaye	Lower Bound:   1 ✓ Navigable Role Exclude Metacla Metaclass	Add Remove	-1

### Figure 12.11. Create Relationship Type Dialog

• Step 4 - Select on Role A and Role B tabs in the dialog and specify role names, Add... any Metaclasses you wish to specifically Include as possible participants. You can also specifically Exclude metaclasses.

Role A Role B Inheritance	Properties		Lower Bound: 1	🚆 Upper Bound: 📑
Include Metaclasses	Ordered 🔽 Unique	<b>v</b>	Navigable Role Prop	perties
Metaclass	Model Class	T	Metaclass	Model Class
Person <com.metama< td=""><td>People</td><td></td><td></td><td></td></com.metama<>	People			
Add	Remove			Add

Figure 12.12. Role A Panel

Role B Name: Parent	rties	Lower Bound: 1	👻 Upper Bound: 📑
Corde	ered 🔽 Unique 🔽	Navigable Role Prop	serties
Metaclass	Model Class	Metaclass	Model Class
Person <com.metamatrix.me< td=""><td>People</td><td></td><td></td></com.metamatrix.me<>	People		
	) >		

Figure 12.13. Role B Panel

You can also change role properties via the **Role Properties...** button which launches a properties editor dialog.

per	rty	Value
	Info	
	Name	LA Son
	Object URI	
	Opposite Role	Role (Path=/RelationshipsExample/FamilyRelationships.xmi/Son)
	Stereotype	LA Person
	Participants	
	Exclude Types	
	Include Types	Person <com.metamatrix.metamodels.people.person></com.metamatrix.metamodels.people.person>
	Maximum Number	L11 -1
	Minimum Number	Las 1
	Navigable	🖙 true
	Ordered	🖙 false
	Unique	🖙 true
	Type Constraint	
	Constraint	LA

#### Figure 12.14. Role Properties Panel

• Step 5 - Select the Inheritance tab.

Role A Role	B Inheritance Properties		
Supertype:	Dependency		Browse
Subtypes:	Name	Path	
			Find
	1		

### Figure 12.15. Inheritance Panel

• Step 6 - Launch the Select Relationship Type dialog via the Browse... button. Choose a base relationship type that best describes your category of relationship. Click OK.

🤷 Select Relati	ionship Type	in an	
Select a relationsh	ip type		
Built In Ty     Gener     Gener	rpes ric ality fication ralization ndency formation osition ss ration ement estation umption liance nipsExample isionsProject		
	Properties	ОК	Cancel

### Figure 12.16. Relationship Type Selection Dialog

- Step 7 Select the Properties tab and change/set relationship type property values.
- Step 8 Click Finish to create relationship type.

# 12.3. Relationship Navigator

A **Relationship Navigator** view provides a unique way of viewing and navigating your relationships. It can be very useful in understanding associations between complex and visibly unrelated data.

To open the **Relationship Navigator** you can select the **Window > Show View > Relationship Navigator** action (if available) or open the **Show View** dialog via **Window > Show View > Other...** and open the Teiid Designer folder and select **Relationship Navigator**.

You can also quick-launch the navigator by selecting any object in the Section 4.1, "Model Explorer View" or a Model Editor and right-click select the Open in Relationship Navigator action.



### Figure 12.17. Open In Relationship Navigator Action

In the example presented here, the diagram below illustrates the standard role-based relationships defined for a family.



Figure 12.18. Sample Relationships Diagram

This action opens the **Relationship Navigator** in the same panel as the **Properties** and **Description** views.



### Figure 12.19. Relationship Navigator View

The visible detail in the navigator is limited in order to simply navigation. You can, however, show detailed information about the navigation/relationship nodes by hovering your mouse and showing tooltips.



### Figure 12.20. Tool-tip Info in Navigator

At the center of the navigator is the focused node, which represents the base or frame of reference for the displayed navigation. In this sample above, George has a wife, a niece and 5 sons. The relationship diagram shown below illustrates the role based view of these same relationships.

You can navigate to through your relationships by double-clicking an end-node in one of the relationships. For instance, double-click the Son Ron in the upper end-node and the navigator now focuses on Ron and his relationships. You can also right-click and select the Follow Relationships action.



### Figure 12.21. Navigating Relationships

As you navigate, a blue ball will animate the direction of the relationship flow first too the focused object then into the end of the relationship containing the initial focused object.

The navigator also provides toolbar and right-click menu actions to perform various functions in the navigator.

# **User Preferences**

#### Go To: Table of Contents [index.html]

The Teiid Designer provides options or preferences which enable customization of various modeling and UI behaviors. Preferences can be accessed via the Edit > Preferences action on the Main toolbar.



Figure 13.1. Preferences Dialog

# **13.1. Teiid Designer Preferences**

General Teiid Designer preferences include.

- Always open editor without prompting To change/edit a model, it must be opened for editing. Checking this box will automatically open the model in an editor if the user attempts to perform a change in a model. If unchecked, the user will be informed that an editor will be opened before the operation is completed.
- Check and update imports during save occasionally editing a model may add or remove objects in one model that reference objects in another model. Model Imports keep track of these dependencies within each model. A validation error or warning may appear during a build. Checking this box will automatically check and update imports during the save process. This will result in any unneeded imports being removed from the model or any required imports added to the model. If unchecked, no updating of imports will be performed.

Designer	
Specify values for general Designer preferences.	
_ Metadata Modeling	
Always open editor without prompting	
Check and update imports during save	

### Figure 13.2. General Teiid Designer Preferences Panel

### 13.1.1. Diagram Preferences

Several diagram preferences are available to customize your diagrams.

- **Notations** Standard diagram notation for Teiid Designer is based on UML notation. Future releases may include alternate notations.
- **Routers** The relationship link type for Package and Custom diagrams (Foreign Key Primary Key relationships) can be customized. Available options include Orthogonal (default), Direct or Manual (user defined breakpoints).
- Font Settings Select font type, style and size.
- **Background Color Settings** Select a unique background color for each diagram type to help differentiate between types.
- **Model Size** Displaying very large diagram may take a considerably long time. This preference allows users to set an upper limit on the number of objects to display in a diagram. If this limit is exceeded, a warning is displayed to the user and the diagram is not constructed.
- **Relationship Options** UML-type relationships can be customized in a couple of ways. Role Names and Multiplicity labels can be shown or hidden using the check-boxes labeled **Show Role Names** and **Show Multiplicity**.

Diagram 🔶 🗸 🖒 🗸
General settings for Diagrams
Notations
Default Notation: UML
Routers
Default Router Style: Orthogonal
Font Settings
Font Type: Tahoma - Regular - 8 View / Change
Background Color Settings
Custom Diagram: Mapping Diagram:
Custom Relationship Diagram: Dependency Diagram:
Relationship Diagram: Transformation Diagram:
Package Diagram:
Model Size
Warn If Model Size is Larger Than 20000
Relationship Options
🔽 Show Role Names
Show Multiplicity
Restore Defaults Apply

### Figure 13.3. Diagram Preferences Panel

# 13.1.2. Diagram Printing Preferences

Diagram print options are stored as preferences. These can be accessed through this preference page, by right-click on diagram **Page Setup action** 

```
Ð
```

or via the **Modify Diagram Printing Preferences** action

located on the vertical diagram toolbar .

Diagram Printing	⇔ -⇔ -
Diagram Print	
- Orientation	
C Portrait 🙃 Landscape	
Scaling	
G Fit to: One page	
C Fit to: One page high, as many as necessary wide	
C Fit to: One page wide, as many as necessary high	
Adjust to:	
100 😴 % normal size	
Margins	
Тор: 0.0	
Left: 0.0	0.0
Bottom: 0.0	
Page Order	
📀 Over, then down	
C Down, then over	
Restore Defaults	Apply

# Figure 13.4. Diagram Preferences Panel

# **13.1.3. Editor Preferences**

### 13.1.3.1. XML Document Preferences

**XML Document Mapping Preferences** provide ways to customize <u>XML Mapping Diagram</u> and <u>Recursion Editor</u> behavior.

ditor	
XML Document Mapping Settings       Table       Transformation         XML Mapping Diagram Settings         Auto-expand tree when model has fewer than       21       mapping         Auto-expand XML document tree to level:       1         Hide Mapping Class Columns by Default       1	ing classes.
XML Document Mapping         Upper XML Recursion Limit:       10         Remove Duplicate Attributes on Merge	

# Figure 13.5. XML Document Preferences Panel

# 13.1.3.2. Table Editor Preferences

<u>Table Editor</u> **Preferences** provide a way to customize the order and the information content for each model object type.

Editor			<mark>⇔</mark> • ⇔ •
XML Document Mapping So Preferences for a given t for it has been viewed. A viewed again for the defa	ettings Table Transfo able will only be present if After restoring defaults, t ault preferences to becom	rmation the table editor tal he table editor must he visible.	) be
Table	Columns		
Base Tables	Column	Visible	
Columns	Location	Yes	
Foreign Keys	Name	Yes	
Indexes Brimary Keys	Name In Source	Yes	
Friday Keys	System	Yes	
	Cardinality	Yes	
	Supports Update	Yes	
	Materialized	Yes	
	Logical Relations	Yes	
	Object URI	Yes	1.000
	Description	Yes	Up
			- 1
			Down
	-		
	-		
	-		
	1		
	-		
1			
		Bestere Defe	ilte Applu
		Restore Derat	ICS Apply

# Figure 13.6. Table Editor Preferences Panel

# 13.1.3.3. Transformation Editor Preferences

<u>Transformation Editor</u> Preferences provide a way to customize SQL formatting, diagram layout, and default view entity properties.

Editor	<b>⇔</b> • ⇔ •
XML Document Mapping Settings Table Transformation	
SQL Clauses	
▼ Start Clauses on New Line ▼ Indent Clause Content	
Diagram Layout	
• View and Sources	
C Tree Layout	
Miscellaneous	
Default String Length: 10	
Restore Defau	lts Apply

### Figure 13.7. Transformation Editor Preferences Panel

# **13.1.4. Validation Preferences**

**Validation Preferences** provide a way to customize the severity of some of the rules checked during model validation.

Validation preference pages, shown below, include Core, Data Access, Relational Model, XML-related and XML Schema (XSD) models.

Validation	
Specify the Severity Level for validation-related occurrences.	
DataAccess Relational XML XSD Core	
DataAccess Validation Control Settings	
Missing length property on elements with string or character types:	Warning 💌
Missing precision property on elements with numeric types:	Warning 💌
Elements of builtin type integer:	Warning 💌
Name in source conflicts of sibling entities:	Ignore 💌
Indexes with columns from more than one table:	Warning 💌

Figure 13.8. Data Access Model Validation Preferences Panel

Validation	
Specify the Severity Level for validation-related occurrences.	
DataAccess Relational XML XSD Core	
Relational Validation Control Settings	
Missing length property on columns with string or character types:	Warning 💌
Missing precision property on columns with numeric types:	Warning 💌
Columns of builtin type integer:	Warning 💌
Name in source conflicts with sibling entities:	Ignore 💌
Indexes with columns from more than one table:	Warning 💌

Figure 13.9. Relational Model Validation Preferences Panel

Validation	
Specify the Severity Level for validation-related occurrences.	
DataAccess Relational XML XSD Core	
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 💌
Mapped XML Elements/Attributes Nillable:	Ignore 💌
Incompatible Datatypes for Column-to-Element/Attribute Mappings:	Warning 💌

# Figure 13.10. XML Document Model Validation Preferences Panel

Validation	\$ • \$ +
Specify the Severity Level for validation-related occurrences.	
DataAccess Relational XML XSD Core	
XSD Validation Control Settings XML Schema Document validation problems	Ignore 💌

# Figure 13.11. XSD Schema Model Validation Preferences Panel

Validation	+
Specify the Severity Level for validation-related occurrences.	
DataAccess Relational XML XSD Core	
Core Validation Control Settings	
Multiple EObjects with same uuid.	Ignore 💌
String functions (SUBSTRING, LOCATE, and INSERT) 0 to 1 base check:	Warning 💌

# Figure 13.12. Core Model Validation Preferences Panel



#### Note

Increasing the severity level to error will prevent you from testing your VDB or deploying a web service if violations of that preference are found during validation.

# 13.1.5. XML Schema Source Preferences

XML Schema as Relational Source Model Preferences provide a way to customize how the relational source model is generated from the schema.

XML Schema as Relational Source Model	↔ ↔ ↔
Parameters for fine tuning the modeling process	
Maximum value of maxOccurs above which an element is not copied in its parent	4
Number of different parents above which an element is not merged into its parent	3
Maximum number of fields above which an element is not copied in its parent	10
Name of request table	REQUEST
Separator character used when merging child element names into parents	-

### Figure 13.13. XML Schema Source Preferences Panel

# **13.2. SQL Explorer Preferences**

<u>SQL Explorer</u> Preferences provide a way to customize the result preview row count and the actual result row count. If a query plan should be generated each time a query is executed, keep the Generate query plans preference checked. Note: Changes to this preference do not affect existing connections.

SQLExplorer		
Preview - Limit Rows	80	
SQL - Limit Rows	2000	
🔽 AutoCommit		
Commit On Close		
✓ Tables and columns auto	-completing assistance. Use only with fast databa:	se connections
🔽 Generate query plans (re	equires connection restart)	
XML Results Display limit (# o	hars) 100000	

#### Figure 13.14. SQL Explorer Preferences Panel

# 13.2.1. SQL Editor Preferences

<u>SQL Editor</u> Preferences provide a way to customize the font properties of the editor and the separator character used when exporting results to the clipboard.

SQL Editor	
Font Properties Text Font: Courier New-regular-10 Change	
Text Properties	
Table name     Color       Column name     Bold       SQL Keyword     Bold       SQL Single Line Comment     Image: SQL Multi Line Comment	
Preview	
select * from MyTable single line comment /* multi line comment select * end multi line comment*/	
Export to Clipboard	
Separator	
· ;	
CL	
🔿 \t [TAB]	
Export column names	

Figure 13.15. SQL Editor Preferences Panel