TIBCO® Data Virtualization Getting Started Guide

Version 7.0.6

Last Updated: November 20, 2017

Important Information

Preface

Conventions

This document uses the following conventions.

Conventions	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`courier font`	Terminal sessions and information the system displays appear in courier font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note: Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Searching Across Multiple PDF Documents

When you are looking for information in the documentation set, you might want to search across multiple documents. You can use the free Adobe Reader to do this. If the options described are not available in your version of Adobe Reader, please update it.

To search your PDF documents

1. Open Adobe Reader.
2. From the File menu, choose Open. and open any PDF document.
3. From the Edit menu, choose Advanced Search.
4. Under Where would you like to search?, click All PDF Documents in.
5. Click My Documents and choose Browse for Location at the bottom of the drop-down list.
6. Browse to the location of all your PDF files.
7. Enter the search term and click Search.

   Acrobat lists all PDFs in the folder that contain the search string, and the number of occurrences in each.
8. Click the instance of the search term and its surrounding text to open the PDF to that page.

Document Change History

This section provides the revision history for the *TDV Getting Started Guide*.

Version Number	Issue Date	Status	Reason for Change
7.0	November 2014		Major software revision. ■ New configuration parameters. ■ Changes in overall product function. ■ Removal of legacy features. ■ Additional features used to help convert legacy resources. ■ Updated drivers and integration for use with TDV. ■ New web based Deployment Manager feature. ■ Revised PAM content. ■ Additional logging features. ■ Removal of support for iPlanet directory service.
7.0.1	March 2015		■ Added updated product architecture.
7.0.2	August 2015		No substantive changes.
7.0.3	December 2015		No substantive changes.
7.0.4	August 2016		No substantive changes.
7.0.5	January 2017		Fixed port numbers.

Introduction

The TIBCO® Data Virtualization (TDV) forms the core of the Data Virtualization Platform.

At build time, developers use the TIBCO Data Virtualization's easy-to-use development environment, Studio, with automated code generators, to create high-quality, semantically meaningful, standards-compliant views and data services. Rich tools enable complex federation and transformation functions. Standard adapters simplify access and publication development activities. The Manager controls features including security, metadata, and source code.

At run time, the TIBCO Data Virtualization's query engine securely queries, accesses, federates, abstracts and delivers data to consuming business solutions on demand. Multiple caching options provide additional speed and flexibility.

Topics covered in this topic:

- Overview, page 5
- TIBCO Data Virtualization Architecture, page 6
- TDV Resources, page 7
- Studio Modeling and Publishing, page 8

Overview

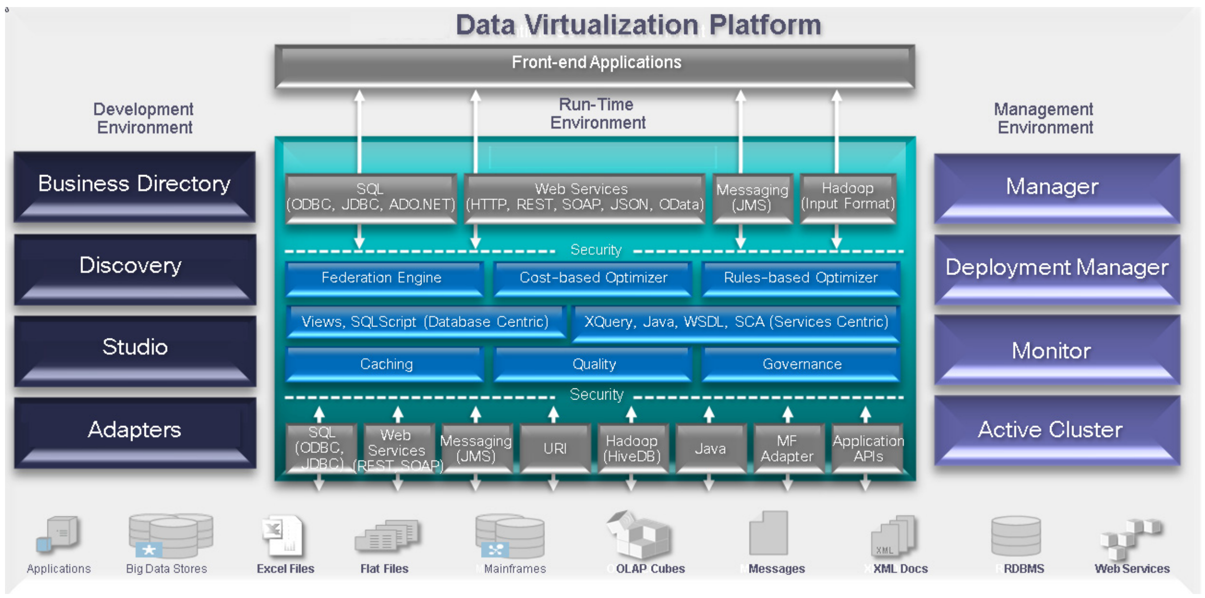
The Data Virtualization Platform is a suite of solutions that enables the definition of a virtual data layer to facilitate discovery, integration, and federation of disparate, distributed information sources. The TDV Server enables creation of a transparent, real-time interface to business information for business users and application developers. Designers using TDV create a securely managed unified view across files, databases, and packaged applications.

TDV supports a wide range of data sources including Oracle, Microsoft SQL Server, MySQL, DB2, Sybase, Informix, Netezza, Microsoft Access, Microsoft Excel, LDAP, flat files (including data in XML), and web services.

The TIBCO Data Virtualization virtual data layer enables client applications to browse, query, update, and manage information gathered from across the enterprise and from any number or type of data sources.

TIBCO Data Virtualization Architecture

The TIBCO Data Virtualization is a data virtualization server that connects to existing data, federates disparate data, abstracts complex data, and delivers the information as data services. The server includes a graphical development environment that lets you rapidly design and develop database-centric objects, including relational views and service-oriented objects. The TIBCO Data Virtualization also includes a complete set of management capabilities.



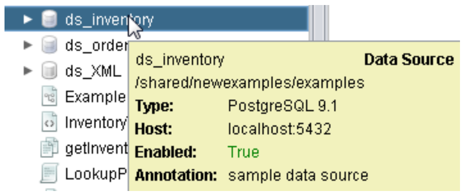
Product	Description	For More Information
TIBCO Data Virtualization	TDV Server represents the core runtime environment that hosts various components such as the query engine and metadata repository.	*TDV User Guide*
Business Directory	Business Directory is a business-friendly interface that provides a catalog of the published resources contained in one or more instances of TDV. Business Directory facilitates collaboration across groups within an organization by supporting communication and distribution of data, and supports the governance of data as it moves through its lifecycle.	*Business Directory Guide*
Discovery	Discovery enables IT professionals to go beyond profiling to examine data, locate key entities and reveal hidden relationships in their enterprise data. You can use that knowledge to build rich data models for data virtualization and other information initiatives. The models allow you to access and show live data, making it easier to validate business requirements with users.	*Discovery User Guide*
Studio	Studio provides an interface for data modeling, querying, transformation, and administration.	*TDV User Guide*

Product	Description	For More Information
Adapters	Adapters accelerate virtualized access to popular enterprise applications, relational and multi-dimensional data sources, big data stores, and Web content. The adapters optimize performance and development. In addition to intelligently evaluating and leveraging underlying data source capabilities to ensure optimal federated query performance, Adapters provide: Certified Vendor-Specific Connectivity API—Access proprietary data using vendor-approved access methods. Relational Representation of Source Data—Standardize data structures to accelerate development. Business Canonical Abstraction of Source Data—Standardize data semantics to simplify development. Vendor-specific Engineered Functions—Provide performance beyond vendors' standard capabilities.	Adapter Guides
Manager	TDV Manager functionality is provided both as a web interface and in Studio. Manager lets you control and monitor the server, users and groups, licensing, security, and other administrative functions.	*Administrator Guide*
Deployment Manager	Deployment Manager is a web-based tool you can use to manage and streamline the development lifecycle of TDV resources. Deployment Manager enables you to build repeatable deployment plans to seamlessly promote resources across environments. Typically, the migration is from development machines to test machines to production machines.	*Administrator Guide*
Monitor	Monitor provides a comprehensive, real-time view of your Data Virtualization platform environment, whether a single TDV instance or a cluster of instances.	*Monitor Guide*
Active Cluster	Active Cluster allows you to scale your TDV deployments and maintain continuous availability of data services. Active Cluster enables you to fulfill service level agreements by increasing capacity on demand, simplifying scaling, and improving manageability.	*Active Cluster Guide*

TDV Resources

TDV resource refers to the resources that are used for data modeling and building business solutions using TDV software. These resources are data sources, views, parameterized queries, SQL scripts, Java procedures, packaged queries, transformations, and TDV data services. Data stored in these resources are available in tabular or hierarchical format, and noted accordingly as either tabular data or hierarchical data.

Let the cursor hover over the name of a resource, and a tooltip displays the resource name, type, parent container, and other useful information such as annotations, status, or target.



ds_inventory Data Source
/shared/newexamples/examples
Type: PostgreSQL 9.1
Host: localhost:5432
Enabled: True
Annotation: sample data source

The parent container path combined with the resource name is the unique identifier for the invocation and reference to any TDV defined resource. For example you can have two tables, but because the parent container path is different and because the name and path used to refer to the resource are case-sensitive it is unique:

```
/shared/examples/ds_inventory/tutorial/inventorytransactions
/shared/production/ds_inventory/tutorial/InventoryTransactions
```

Resources include the following:

- Data sources
- Views
- Procedures
- Definition sets
- Triggers
- TDV data services
- Folders
- Tabular and hierarchical data
- Resources in context

For more information, see the *TDV User Guide*.

Studio Modeling and Publishing

Metadata modeling and publishing in TIBCO Data Virtualization is a three-step process where you will:

- Introspect
- Model
- Publish

For details on these three processes, see the *TDV User Guide*.

Introspect

To introspect is to examine a physical data source and selecting only the specified resources from that data source for the sake of modeling in the TDV system. Introspection is a part of connecting to the data source. TDV lets you specify the physical data source you want to examine.

When you examine a data source, you do not have to choose the entire data source but can be selective about the data source resources you want to use for building a solution for your specific business.

Model

To model, you create, design, and edit views and procedures based on the introspected data sources.

You can experiment with different joins, columns, and constraints on views and procedures to find one that generates the result set that meets your business needs and drives your information integration project. You can also experiment with caching and join ordering to find the preferred performance profile, and subsequently store the view or procedure to be scheduled for batch reporting or to publish as a TDV database or Web service.

Publish

To publish you make the specified views and procedures available to client applications in the enterprise.

Logging Into TDV Server

This topic describes how to set up the work environment for using the software after installation.

- Connecting to TDV Server and Starting Studio, page 11
- Studio Modeler and Resource Tree Overview, page 13
- Viewing and Opening Resources, page 13

The installation process installs the server and other selected components in a specified location and starts its repository database.

TDV Server and Studio are available in a program group on the Start menu. TDV Server starts automatically after installation.

Connecting to TDV Server and Starting Studio

To log into TDV Server, you should:

- Obtain a valid username and password from your TDV administrator
- Know the name of the domain (composite or LDAP) to which you belong
- Know the name of the machine or the IP address where TDV Server is installed

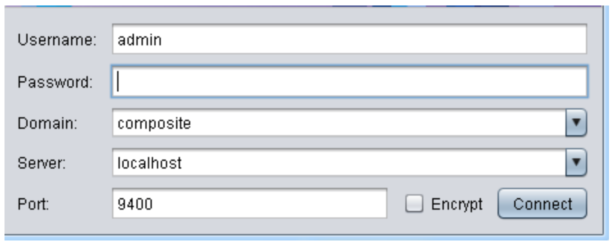
For details on installing the software, see the *TDV Installation and Upgrade Guide*.

This guide provides default values for the tutorial resources and sample data that can be installed with TDV.

To start Studio and log into TDV Server

1. Select the Studio option.

Start > All Programs > tdv> <version> > Studio > Start Studio <version>.



2. Type or select the following sign-in information.

Field	Tutorial Values	Description
Username	admin (default)	Username.
Password	admin	Password associated with your username.
Domain	composite (default)	Your domain from the drop-down list. This entry is used for authentication to connect to the server.
Server	localhost (default)	The host machine for the server in the Server drop-down list. Use the default if Studio and the server are running on the same machine. If not, enter the IP address or the hostname of the machine where the server is installed.
Port	9400 (default)	Specify the HTTP base port number through which you can connect to the server. Change the default value (or current value) if necessary.
Encrypt	unchecked	To connect to the server securely using SSL through HTTPS, check the Encrypt check box; otherwise HTTP clear text is used.

If Kerberos Single Sign-On has been enabled for your network the SSO check box appears and you can check it to use your Windows sign-in to connect with the TIBCO Data Virtualization. Otherwise, proceed to sign in using username and password authentication.

3. Click Connect.

If the log-in credentials are validated successfully, you connect to TDV Server, and Studio opens for your modeling.

When the TDV Server is set to accept HTTPS secure mode communications only, a sign-in attempted without the Encrypt check box results in a refused connection. The sign-in errors look similar to the following:

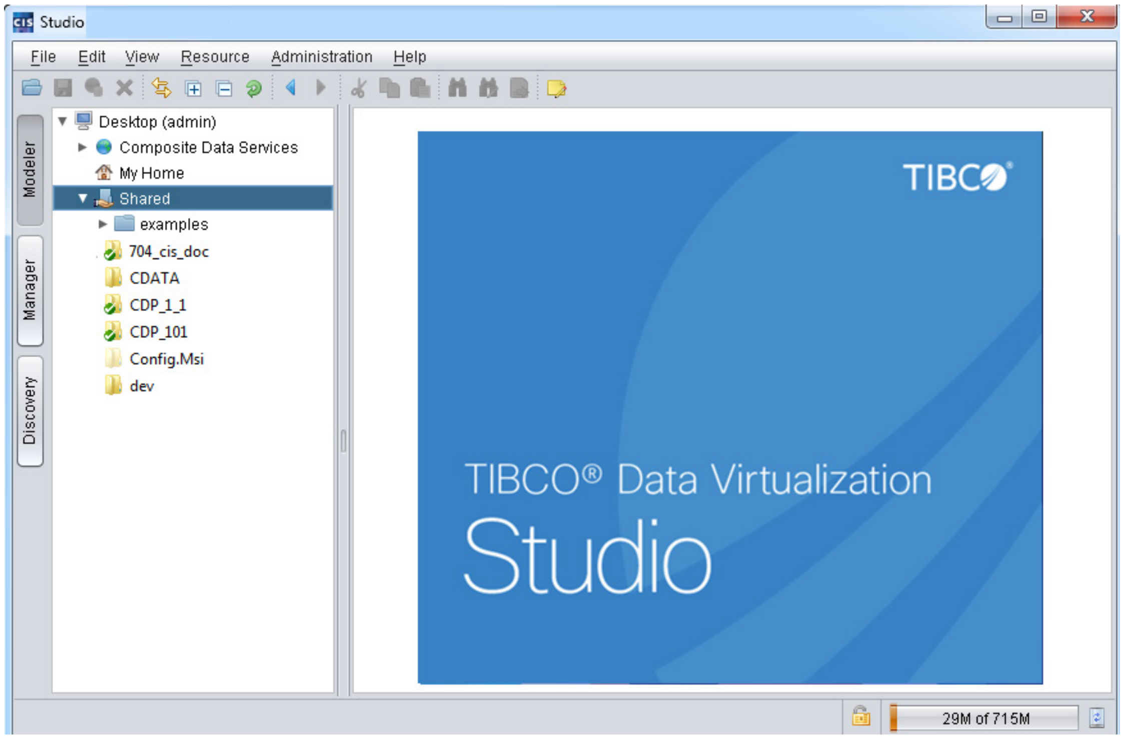
```
Failed to connect to http://localhost:9400/cdms/webapi; nested exception is:java.ami
RemoteException: HTTP transport error. java.net.ConnectException...
```

Dismiss this error and use the Encrypt check box to sign in securely.

Studio Modeler and Resource Tree Overview

When you start Studio, the Modeler in Studio is displayed by default. The Modeler is the area where all data modeling activities take place. The left pane in the Modeler displays all available resources in a tree format, which is referred to as the *resource tree*.

Expand the node labeled "examples" that is inside the Shared folder.



- Desktop—The current user’s virtual work area in TDV Server, and is like the desktop for a personal computer’s user interface.
- Data Services—Resources visible to client applications that connect to TDV Server.
- My Home—The current user’s workspace in TDV Server.

  You can create TDV resources in My Home.
- Shared—Resources shared by all users in the system.

  You can create TDV resources in Shared.
- examples—Sample resources to help you get started using TDV.
- <host machine>—This area reflects the machine that hosts TDV Server.

For further details on the resource tree, see the *TDV User Guide*.

Viewing and Opening Resources

Some resources have been precreated and are displayed in the resource tree. Only those users with administrative privileges can delete these resources. Others can view and execute them.

To view the example resources

1. Expand the Shared > examples folder in the resource tree, and expand each node to see what is available.

   For example, Shared > examples > ds_orders > tutorial displays the tables in the ds_orders data source.

2. Expand the Composite Data Services > Databases > examples folder in the resource tree, and expand each node to see what is available. Items might not be viewable here until after you complete Getting Started with TDV, page 15.

To open a resource

1. Right-click the resource, and select Open.

   In the case of a leaf-node, you can also double-click it.

Getting Started with TDV

This topic provides a step-by-step tutorial that describes how to use the Modeler in Studio to create TDV data resources.

Topics covered in this topic include:

- About this Tutorial, page 15
- Adding Data Sources, page 16
- Creating the XML Data Transformation, page 23
- Building Simple Views, page 25
- Creating a Composite View, page 32
- Publishing Your Views, page 33
- What You Learned in this Tutorial, page 34

About this Tutorial

This tutorial teaches you how to build a unified solution to address a typical business scenario. For this example, you represent the three departments at ALPHA and build a unified view of your distributed business systems to address customer-reported issues. This example shows how to build the unified view using the following sequence of tasks.

Task	Description	Instructions
1	Add data sources to TDV. You will add the three underlying data sources to the TDV metadata repository so you can query them. Each of these data sources will display as a data source in the resource tree.	Adding Data Sources, page 16
2	Build three simple views to retrieve data from the underlying data sources. Each of these views will extract specific information for you to address the business issue.	Building Simple Views, page 25
3	Join tables, and provide aliases for column names.	
4	Create the final unified view by combining the three individual views. This view unifies the separate pieces of information you retrieve through the three individual views you create.	Creating a Composite View, page 32
5	Create a TDV database. This database will be visible to client applications using JDBC and/or ODBC to connect to TDV Server.	Publishing Your Views, page 33
6	Publish the unified views as TDV database tables. This unified view provides a single view solution for the business issue.	Publishing Your Views, page 33

For a quick tour of how to create and publish a REST service, see the tutorials in the *TDV User Guide*.

Adding Data Sources

Adding a data source means creating a TDV representation of the actual, underlying data source in the TDV metadata environment for data modeling. In some cases, you might add the entire data source and in others you might be selective about which resources in the data source that you want to add. The TDV representation of the data source consists of two things: the metadata that defines the data source and the information about connecting the data source to TDV Server.

Provided you have the WRITE privilege, you can add a data source to any area except Data Services in the resource tree.

For details on privileges, see the *TDV Administration Guide*.

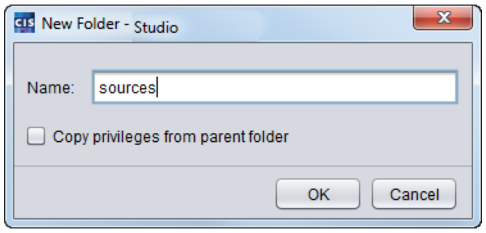
- Creating a Folder, page 16
- Adding the Orders Data Source, page 16
- Viewing the Data Source Schema, page 19
- Adding the Inventory Data Source, page 20
- Adding the XML Data Source, page 21

Creating a Folder

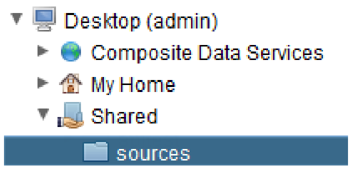
To create a folder named sources

1. Start Studio, if it is not currently running using the instructions in Connecting to TDV Server and Starting Studio, page 11.
2. Right-click Shared, and select New Folder.

   The New Folder window opens for you to name the new folder.
3. Type sources for the name in the input field, and click OK.



The newly created folder is added to the resource tree under Shared.



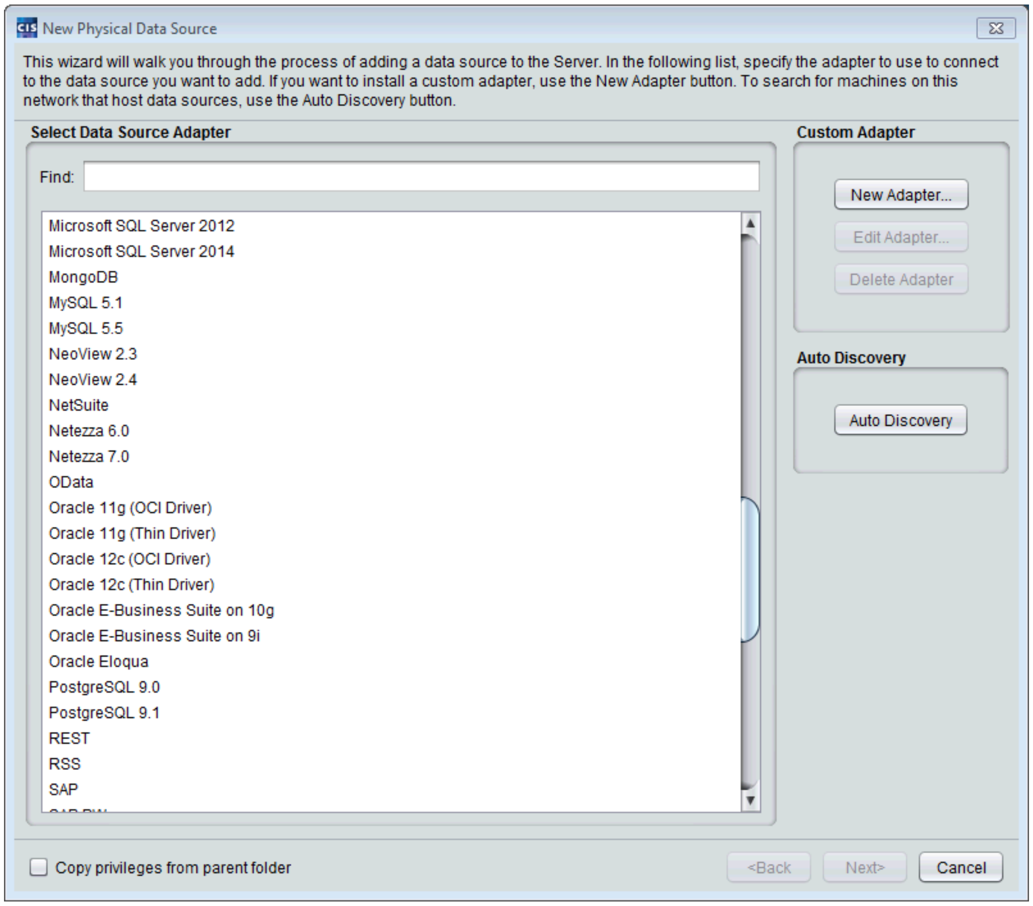
Adding the Orders Data Source

In this section you add the orders data source.

To add the orders data source

1. Right-click the sources folder and select New Data Source.

2. Scroll down and select PostgreSQL 9.1 as the data source adapter.



This wizard will walk you through the process of adding a data source to the Server. In the following list, specify the adapter to use to connect to the data source you want to add. If you want to install a custom adapter, use the New Adapter button. To search for machines on this network that host data sources, use the Auto Discovery button.

Microsoft SQL Server 2012
Microsoft SQL Server 2014
MongoDB
MySQL 5.1
MySQL 5.5
NeoView 2.3
NeoView 2.4
NetSuite
Netezza 6.0
Netezza 7.0
OData
Oracle 11g (OCI Driver)
Oracle 11g (Thin Driver)
Oracle 12c (OCI Driver)
Oracle 12c (Thin Driver)
Oracle E-Business Suite on 10g
Oracle E-Business Suite on 9i
Oracle Eloqua
PostgreSQL 9.0
PostgreSQL 9.1
REST
RSS
SAP

3. Click Next.

   The window for specifying the data source information appears.

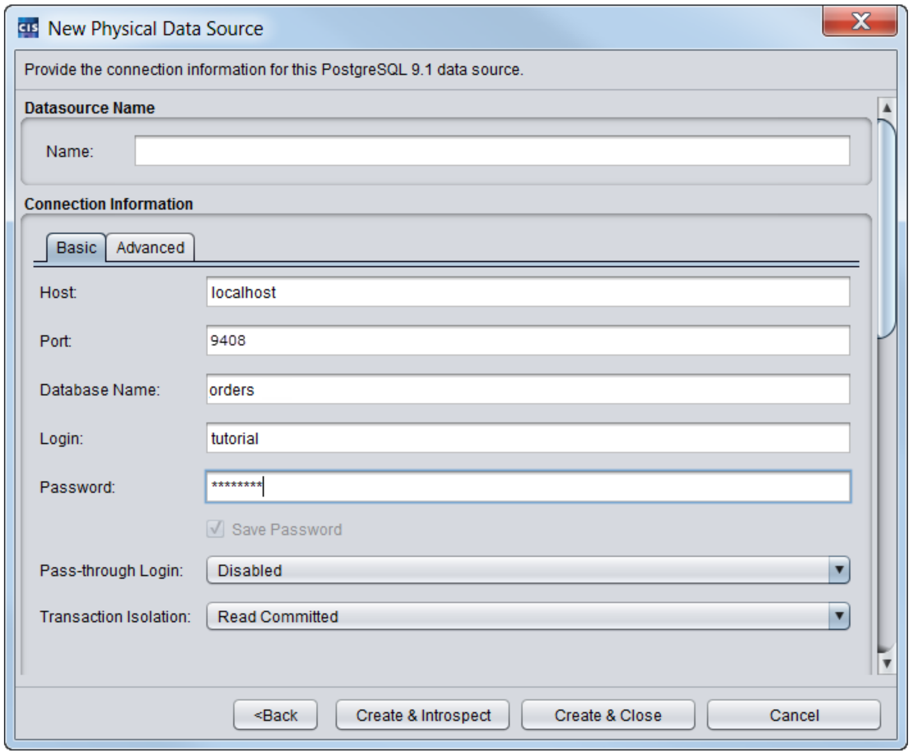
4. Select the Basic tab if it is not already selected.

5. Specify values for the fields shown in the table.

Field Name	Tutorial Value	Description
Name	ds_orders	Type a unique name for the data source. When the process of adding this data source is complete, the name ds_orders will be displayed in the resource tree. ds_orders is the TDV representation of the orders database.
Host	localhost	Type the name or IP address of the machine where TDV Server is installed. If the server is installed on your local machine, you can type the machine name or the term localhost.
Port	9408	Type the port number. Use 9408 for the tutorial to access the server that is shipped with the TDV software. If you use an external server, the default port might be different.

Field Name	Tutorial Value	Description
Database Name	orders	TDV Server uses this entry to locate the orders database instance.
Login	tutorial	These entries are the username and password to access the underlying data source.
Password	tutorial	

For details on the fields and the Advanced tab, see the *TDV User Guide*.



Provide the connection information for this PostgreSQL 9.1 data source.

Datasource Name

Name:

Connection Information

Host: localhost

Port: 9408

Database Name: orders

Login: tutorial

Password: ********

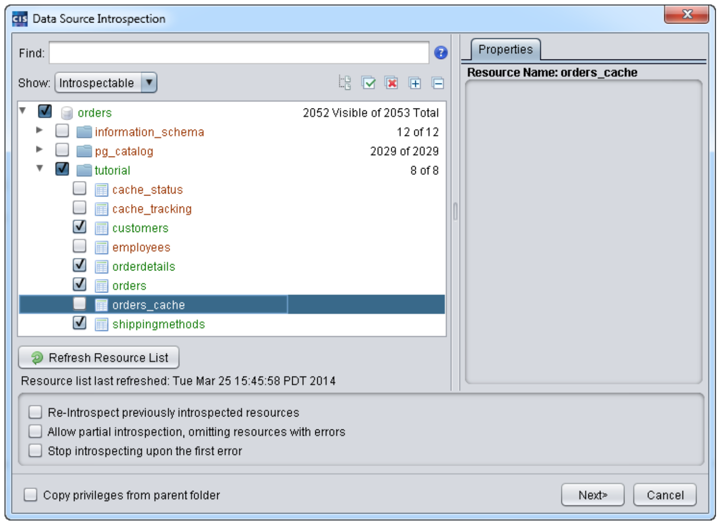
✓ Save Password

Pass-through Login: Disabled

Transaction Isolation: Read Committed

6. Click Create & Introspect.
7. Expand **tutorial**.
8. Select customers, orderdetails, orders, and shippingmethods.

These tables contain the information you need for customer contact and order details.



Data Source Introspection

Find:

Show: Introspectable

orders	2052 Visible of 2053 Total
information_schema	12 of 12
pg_catalog	2029 of 2029
tutorial	8 of 8
☐ cache_status	
☐ cache_tracking	
✓ customers	
☐ employees	
✓ orderdetails	
✓ orders	
☐ orders_cache	
✓ shippingmethods	

Refresh Resource List

Resource list last refreshed: Tue Mar 25 15:45:58 PDT 2014

☐ Re-Introspect previously introspected resources

☐ Allow partial introspection, omitting resources with errors

☐ Stop introspecting upon the first error

☐ Copy privileges from parent folder

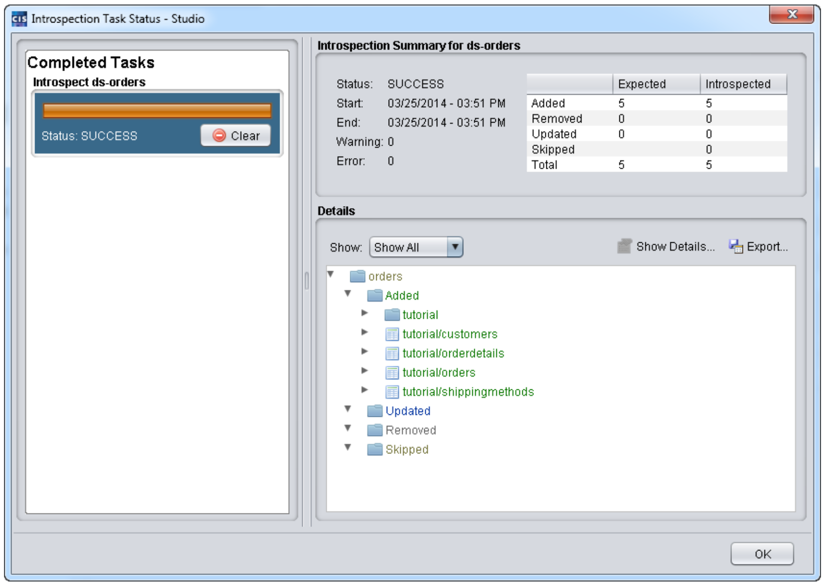
Next> Cancel

Properties

Resource Name: orders_cache

9. Click Next.

10. Click Finish.



Introspection Task Status - Studio

Completed Tasks

Introspect ds-orders

Status: SUCCESS Clear

Introspection Summary for ds-orders

Status: SUCCESS
Start: 03/25/2014 - 03:51 PM
End: 03/25/2014 - 03:51 PM
Warning: 0
Error: 0

	Expected	Introspected
Added	5	5
Removed	0	0
Updated	0	0
Skipped		0
Total	5	5

Details

Show: Show All Show Details... Export...

- orders
  - Added
    - tutorial
    - tutorial/customers
    - tutorial/orderdetails
    - tutorial/orders
    - tutorial/shippingmethods
  - Updated
  - Removed
  - Skipped

OK

11. Click OK.

A folder named ds_orders now appears under sources in the resource tree.

Viewing the Data Source Schema

In this section you view the data source schema.

To view the data source schema (optional)

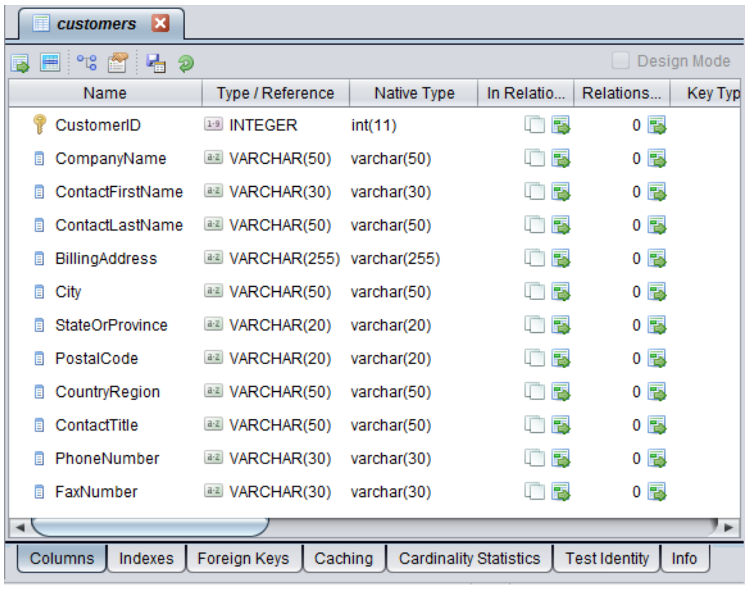
1. Expand Shared > examples > ds_orders > tutorial.

Each node within ds_orders>tutorial represents a table. All of these tables are available for your modeling in Studio.

2. Double-click the customers table name.

   The table editor opens on the right. TDV data type and native data types are shown for each of the tables columns.

   You will see several columns of information. If you run a Discovery process you can discover if a column has relationships or potential relationships to other data sources. From this information you can more effectively model your data environment.



Name	Type / Reference	Native Type	In Relatio...	Relations...	Key Typ
CustomerID	INTEGER	int(11)		0	
CompanyName	VARCHAR(50)	varchar(50)		0	
ContactFirstName	VARCHAR(30)	varchar(30)		0	
ContactLastName	VARCHAR(50)	varchar(50)		0	
BillingAddress	VARCHAR(255)	varchar(255)		0	
City	VARCHAR(50)	varchar(50)		0	
StateOrProvince	VARCHAR(20)	varchar(20)		0	
PostalCode	VARCHAR(20)	varchar(20)		0	
CountryRegion	VARCHAR(50)	varchar(50)		0	
ContactTitle	VARCHAR(50)	varchar(50)		0	
PhoneNumber	VARCHAR(30)	varchar(30)		0	
FaxNumber	VARCHAR(30)	varchar(30)		0	

Adding the Inventory Data Source

To add the inventory data source

1. Right-click the sources folder
2. Select New Data Source.
3. Scroll down, select PostgreSQL 9.1 as the data source adapter, and click **Next**.
4. Type values for the fields shown in the table.

Field Name	Tutorial Value	Description
Name	ds_inventory	Type a unique name for the data source. When the process of adding this data source is complete, the name ds_inventory will be displayed in the resource tree. ds_inventory is the TDV representation of the underlying inventory data source.
Host	localhost	Type the name or IP address of the machine where TDV Server is installed. If the server is installed on your local machine, you can type the machine name or the term localhost.
Port	9408	Type the port number. Use 9408 for the tutorial to access the server that is shipped with the TDV software. If you use an external server, the default port might be different.

Field Name	Tutorial Value	Description
Database Name	inventory	TDV Server uses this entry to locate the underlying data source orders.
Login	tutorial	These entries are the username and password to access the underlying data source.
Password	tutorial	

5. Click Create & Introspect.
6. Expand tutorial.
7. Select the following tables:
   - inventorytransactions
   - products
   - purchaseorders
   - suppliers

   These tables contain the information about product inventory transactions, purchase order details, and supplier contact information.
8. Click Next.
9. Click Finish.
10. Click OK.

    A folder named ds_inventory now appears under sources in the resource tree.

    If you do not see the newly added data source in the resource tree, right-click the Desktop node, select Refresh, and look for it in sources.
11. Expand Shared > sources > tutorial > ds_inventory. Each node within ds_inventory represents a table you selected. Expand each node to see the columns in that table.

    All of these tables and columns are available for your modeling in TDV.
12. If you want to view the schema of this data source, follow the steps described for viewing the orders data source schema in Viewing the Data Source Schema, page 19.

Adding the XML Data Source

The name of the XML data source used here is productCatalog.xml. It is located in the installation directory where the TDV software is installed. By default, on a Windows computer this location is:

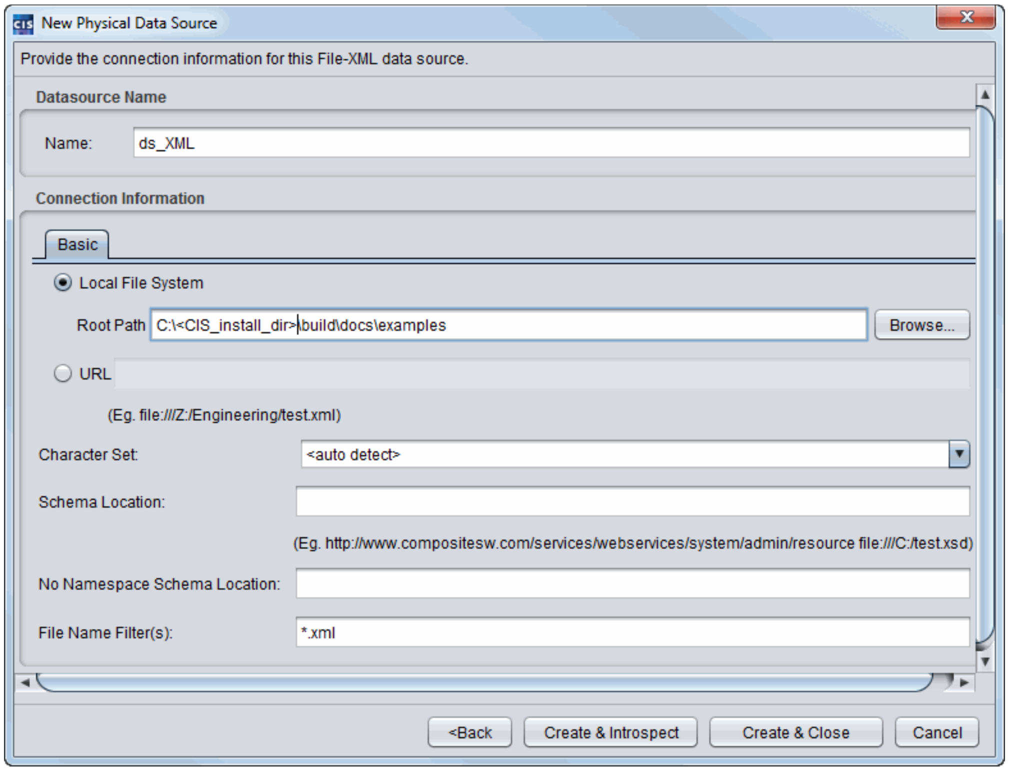
```
<TDV_install_dir>\docs\examples
```

To add the XML data source

1. Navigate to Shared > sources in the resource tree, right-click sources, and select New Data Source.
2. Select File-XML as the data source adapter type, and click Next.

   TDV displays the dialog for you to provide the connection information for this type of data source.

3. In the New Physical Data Source window, type values for the fields on the Basic tab.



Provide the connection information for this File-XML data source.

Name: ds_XML

● Local File System

Root Path C:\<CIS_install_dir>\build\docs\examples

○ URL

(Eg. file:///Z:/Engineering/test.xml)

Character Set: <auto detect>

Schema Location:

(Eg. http://www.compositesw.com/services/webservices/system/admin/resource file:///C:/test.xsd)

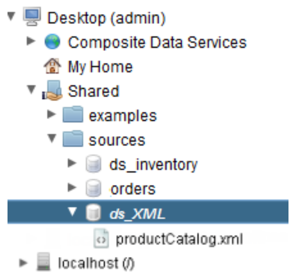
No Namespace Schema Location:

File Name Filter(s): *.xml

Field name	Tutorial Value	Description
Name	ds_XML	Type a unique name for the data source. This name is user-defined and will display in the resource tree when the process of adding the data source is complete.
Local File System	<TDV_install_dir>\docs \examples	Select the Local File System radio button and use the Browse button to locate the root path to the XML file, as in the following example: C:\Program Files\tdv\7.0\docs\examples The root path does not include the name of the XML file, and only points to the directory of the file. You can also type the root path to the XML file in the Root Path field, instead of using the Browse button.
File Name Filter(s)	*.xml	

4. Click Create & Introspect.
5. Select productCatalog.xml.
6. Click Next.
7. Click Finish.
8. Click OK.

The ds_XML data source and associated schema file is added to the resource tree at Shared > sources.



Creating the XML Data Transformation

Transforming data is a common practice when manipulating data within the virtual data environment. For more details on transformation, see the *TDV User Guide*.

The inventory and orders data sources contain tabular data (that is, relational tables), so they are ready to be queried in TDV's modeling environment. But the XML data source (produtCatalog.xml) contains hierarchical data in XML format, so it must be transformed (or, flattened) into a table so you can use the data in your modeling.

For a working example of the transform that we are recreating with these steps, you can open Examples >productCatelog_xform. The steps to recreate the transform include:

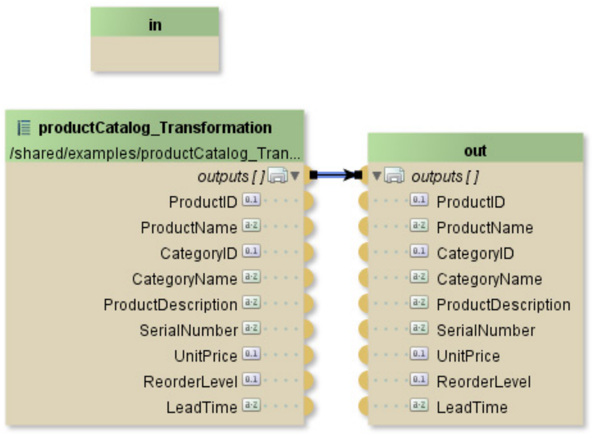
- Creating the Transformation Container, page 23
- Adding Loop and Cast Functions, page 24

Creating the Transformation Container

1. Navigate to Shared > sources in the resource tree, right-click sources, and select New Transformation.
2. Select Any-Any Transform.
3. Click **Next**.
4. In the Transformation Name field, type productCatelog_xform. This should always be a unique name.
5. Click Finish.

   The transformation is added to the resource tree under Shared > sources and the Transformation Editor opens.
6. Expand Shared > examples.
7. Drag the legacy style productCatelog_transform onto the editor.

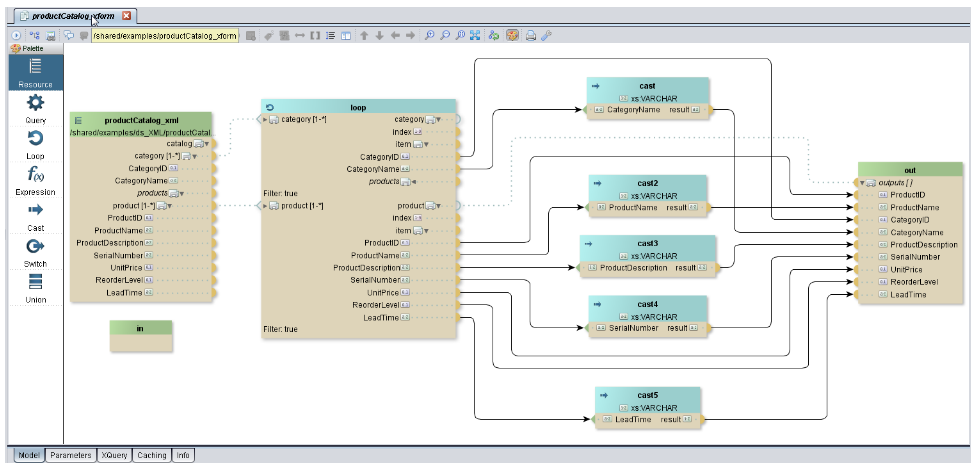
8. Connect it to the out operation. So your transform should look similar to:



9. Delete productCatelog_transform. The out operation should retain all the necessary outputs and their data types.
10. Drag productCatalog.xml onto the editor from Shared > sources > ds_XML.
11. Save your transform.

Adding Loop and Cast Functions

The goal for these steps is to end up with a transform that looks similar to the following:



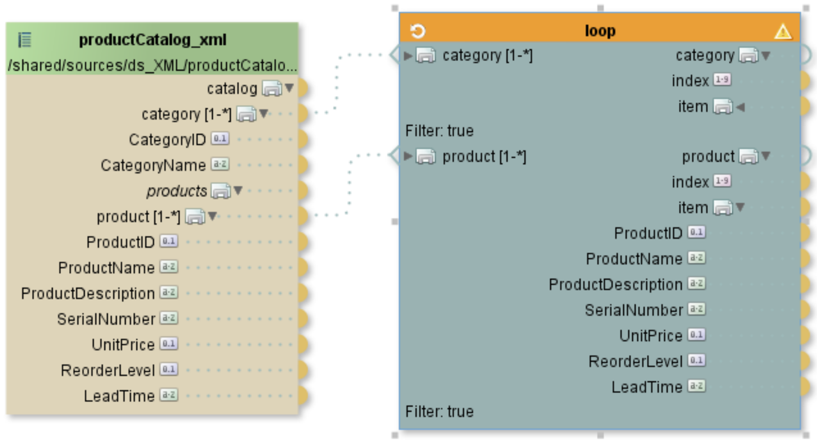
To add loop and cast operations to your transform

1. Expand the hierarchy of the elements within the productCatelog_xml object.
2. Click the Loop icon. It typically looks like:

3. Click in the Transformation Editor model.

4. Click the handle next to **category [1-*]** in productCatelog_xml and drag a line across to the source on the left side of the loop object.
5. Click the handle next to **product [1-*]** in productCatelog_xml and drag a line across to the loop object.

   You should now have something similar to:



6. Connect products in the loop operation to outputs in the out operation.
7. Expand category [1-*] and item in the loop operation.
8. Connect the following Loop elements to the following cast operations:

Loop Element	Out Operation Name	Insert Cast
ProductID	ProductID	
ProductName	ProductName	Click yes at the popup.
ProductDescription	ProductDescription	Click yes at the popup.
SerialNumber	SerialNumber	Click yes at the popup.
LeadTime	LeadTime	Click yes at the popup.
CategoryID	CategoryID	
CategoryName	CategoryName	Click yes at the popup.
UnitPrice	UnitPrice	
ReorderLevel	ReorderLevel	

9. Save and close the transform.

You will query these sources to investigate the issue for ALPHA, to find out why its customer, Landmark Systems, did not receive ALPHA’s product Widget 5 on the date promised (2/10/03). The next step is to build and execute views to obtain specific information about the activities in the Order, Purchase, and Sales departments at ALPHA.

Building Simple Views

This section describes how to create folders for organizing and storing your views, and how to create different views to explore the activities of the departments at ALPHA.

Building a view gives you a model, and executing it queries the relevant data sources and retrieves the specified data. For additional information on building and executing views, see the *TDV User Guide*.

In this section, you create the following three views to describe the data from the three departments at ALPHA:

- Building and Executing the Order Information View, page 26
- Building and Executing the Supplier Information View, page 29
- Building and Executing the Sales Information View, page 30

Building and Executing the Order Information View

To obtain information on the order and customer, you use the ds_orders data source which contains:

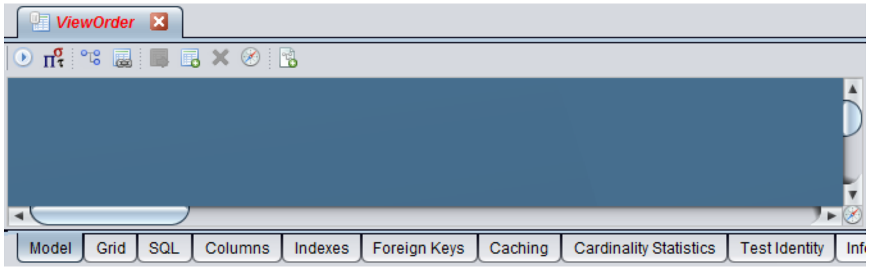
- Order information in the orders table.
- Details of each order in the orderdetails table.
- Customer information in the customers table.

You will include the tables orders, orderdetails, and customers in this view.

To build and execute the order information view

1. Right-click Shared > sources.
2. Select New View.
3. Type ViewOrder as the name for the view.
4. Click OK.

   When this view is added to the folder, the view editor opens for your use in the right pane of the Modeler.



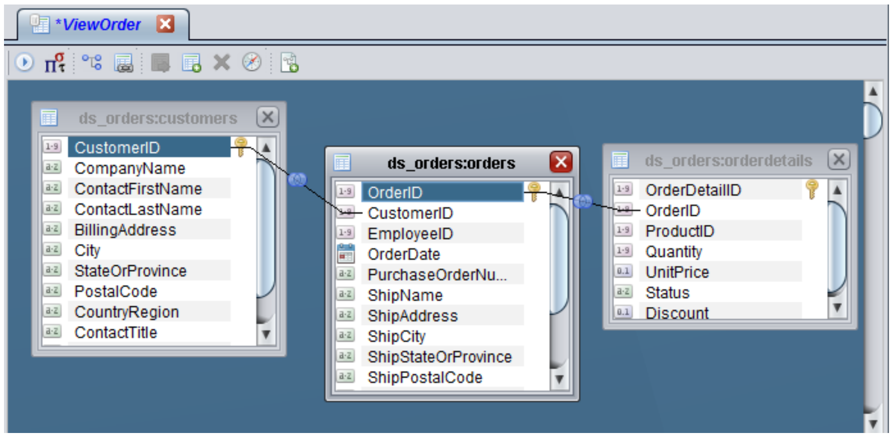
   The editor has the following tabs:

Tab	Description
Model	Use to assemble the tables.
Grid	Use to specify query constraints and the columns to include in the output when you execute your view.
SQL	Use to display the SQL for the view when you design the view in the Model and Grid panels. You can also type SQL in the SQL tab.
Columns	Use to list the columns to select for projection in the view execution result.
Indexes	Use to view reports on the index in the system tables, if the view is published.
Foreign Keys	Use to define foreign keys.
Caching	Use to access the cache setting panel to configure the caching.

Tab	Description
Cardinality Statistics	This panel provides an entry point to access the statistics setting panel where you can configure the statistics for the cached view.
Test Identity	This panel is provided for the row-based security feature.
Info	Use to provide annotations on the view.

5. From the ds_orders data source drag the following tables into the Model panel in the editor:
   - customers
   - orderdetails
   - orders
6. Click OrderID in orders and drag it onto OrderID in orderdetails.

   A line appears representing the inner join between the two tables.
7. Join CustomerID in customers with CustomerID in orders.



   For details on joins, see the *TDV User Guide*.
8. Select the Grid panel.

   The asterisk in the first cell under Column indicates that all the columns in all the tables are selected for retrieval in the result set when the view is executed.
9. To limit the columns in the result set:
   a. Click the first cell under Column.
   b. Select orderdetails.orderid in the drop-down list.
   c. Click more rows and select the following columns as you did in the preceding step:
   - orderdetails.status
   - orderdetails.productid
   - orderdetails.discount
   - orders.orderdate

- customers.companyname
- customers.contactfirstname
- customers.contactlastname
- customers.phonenumber

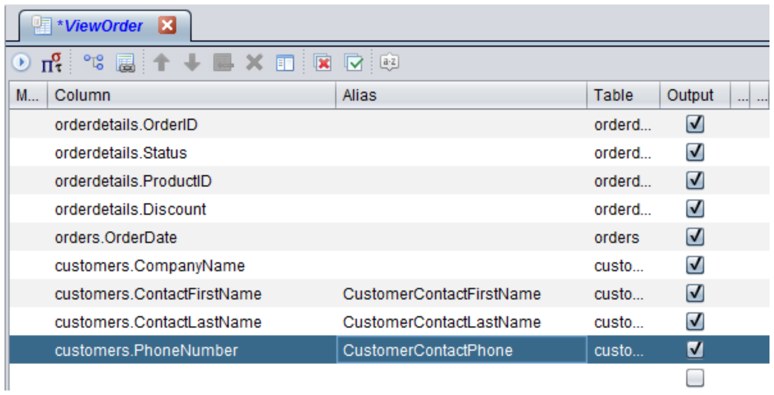
10. In the Alias column, click the cell next to customers.contactfirstname.
11. Type the alias CustomerContactFirstName.
12. Press the Enter key after typing the alias.

    An alias makes a column name unique, which avoids conflicts with columns from other data sources having the same name.

13. Type the alias for each of the following columns.

Column	Alias to type
customers.contactlastname	CustomerContactLastName
customers.phonenumber	CustomerContactPhone

The asterisks next to the View Order indicates that you have made changes but have not saved them.



M...	Column	Alias	Table	Output	...
	orderdetails.OrderID		orderd...	✓	
	orderdetails.Status		orderd...	✓	
	orderdetails.ProductID		orderd...	✓	
	orderdetails.Discount		orderd...	✓	
	orders.OrderDate		orders	✓	
	customers.CompanyName		custo...	✓	
	customers.ContactFirstName	CustomerContactFirstName	custo...	✓	
	customers.ContactLastName	CustomerContactLastName	custo...	✓	
	customers.PhoneNumber	CustomerContactPhone	custo...	✓	
				☐	

14. Save the view.
15. Optionally, you can view the SQL for this view by selecting the SQL tab. After you view the SQL, return to the modeling area by selecting the Model tab.

    Typing or editing the SQL in the SQL panel invalidates the design made in the Model and Grid panels.

16. Execute the view by clicking the Execute button.

    The Result panel opens and displays the result of the view's SQL execution.

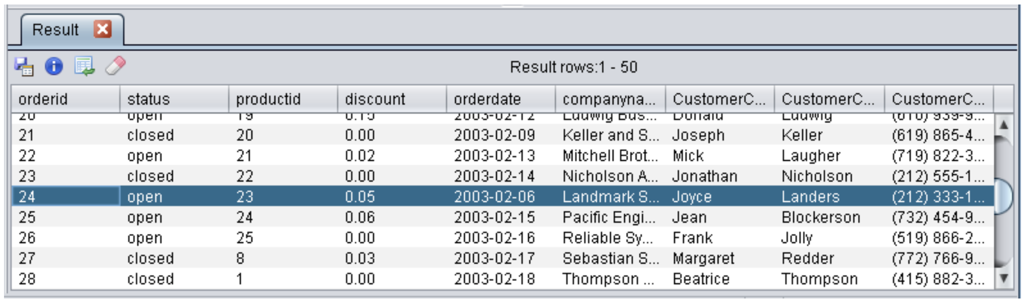
17. In the result displayed in the Result panel, identify the row for OrderID = 24, which has the following data:

```
OrderID: 24
Status: open
ProductID: 23
Discount: 0.05
OrderDate: 2003-02-06
CompanyName: Landmark Systems
CustomerContactFirstName: Joyce
CustomerContactLastName: Landers
```

```
CustomerPhone: (212) 333-1000
```

This result set contains the basic information for the Order department about the order identification, order date, product identification, and customer that are all relevant for the current example.

The following screen shows the view execution results displayed in the Result panel.



orderid	status	productid	discount	orderdate	companyna...	CustomerC...	CustomerC...	CustomerC...
20	open	19	0.15	2003-02-12	Ludwig Bus...	Donald	Ludwig	(610) 939-9...
21	closed	20	0.00	2003-02-09	Keller and S...	Joseph	Keller	(619) 865-4...
22	open	21	0.02	2003-02-13	Mitchell Brot...	Mick	Laugher	(719) 822-3...
23	closed	22	0.00	2003-02-14	Nicholson A...	Jonathan	Nicholson	(212) 555-1...
24	open	23	0.05	2003-02-06	Landmark S...	Joyce	Landers	(212) 333-1...
25	open	24	0.06	2003-02-15	Pacific Engi...	Jean	Blockerson	(732) 454-9...
26	open	25	0.00	2003-02-16	Reliable Sy...	Frank	Jolly	(519) 866-2...
27	closed	8	0.03	2003-02-17	Sebastian S...	Margaret	Redder	(772) 766-9...
28	closed	1	0.00	2003-02-18	Thompson ...	Beatrice	Thompson	(415) 882-3...

18. Close the ViewOrder view tab.

Building and Executing the Supplier Information View

To obtain information on the purchase order and supplier, you use this information in the ds_inventory data source:

- Product transaction information in the inventorytransactions table.
- Purchase details in the purchaseorders table.
- Supplier information in the suppliers table.

To build and execute the supplier information view

1. Right-click Shared > sources in the resource tree.
2. Create a new view named ViewSupplier.
3. Drag the following tables from ds_inventory and drop them into the Model panel of ViewSupplier:
   - inventorytransactions
   - purchaseorders
   - suppliers
4. Join the following columns to each other.

Table Name	Column Name	Join	Table Name	Column Name
suppliers	supplierid	to	purchaseorders	supplierid
purchaseorders	purchaseorderid	to	inventorytransactions	purchaseorderid

5. In the Grid panel, limit the query to the following columns as described in Building and Executing the Order Information View, page 26:

```
inventorytransactions.productid
inventorytransactions.transactionid
purchaseorders.daterequired
purchaseorders.datepromised
purchaseorders.shipdate
purchaseorders.supplierid
suppliers.suppliername
suppliers.contactname
suppliers.phonenumber
```

6. Provide aliases for the following column names.

Column Name	Alias to type
suppliers.contactname	suppliercontactname
suppliers.phonenumber	supplierphonenumber

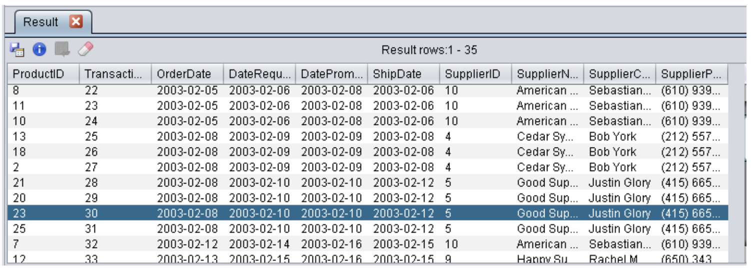
7. Save the view.

8. Execute the view to see the result.

9. In the result set displayed in the Result panel, identify the row for ProductID = 23 and Transaction ID = 30 and which has the following data:

```
ProductID: 23
TransactionID: 30
DateRequired: 2003-02-10
DatePromised: 2003-02-10
ShipDate: 2003-02-12
SupplierName: Good Supplies International
SupplierID: 5
SupplierContactName: Justin Glory
SupplierPhoneNumber: 415-665-8000
```

This result set contains all the information the Purchase department needs about the purchase order and supplier that are relevant to the current example. The supplier did not deliver the order on the promised date of 2003-02-10. Additionally, the supplier shipped the order late (on 2003-02-12).



ProductID	Transacti...	OrderDate	DateRequ...	DateProm...	ShipDate	SupplierID	SupplierN...	SupplierC...	SupplierP...
8	22	2003-02-05	2003-02-06	2003-02-08	2003-02-06	10	American ...	Sebastian...	(610) 939...
11	23	2003-02-05	2003-02-06	2003-02-08	2003-02-06	10	American ...	Sebastian...	(610) 939...
10	24	2003-02-05	2003-02-06	2003-02-08	2003-02-06	10	American ...	Sebastian...	(610) 939...
13	25	2003-02-08	2003-02-09	2003-02-09	2003-02-08	4	Cedar Sy...	Bob York	(212) 557...
18	26	2003-02-08	2003-02-09	2003-02-09	2003-02-08	4	Cedar Sy...	Bob York	(212) 557...
2	27	2003-02-08	2003-02-09	2003-02-09	2003-02-08	4	Cedar Sy...	Bob York	(212) 557...
21	28	2003-02-08	2003-02-10	2003-02-10	2003-02-12	5	Good Sup...	Justin Glory	(415) 665...
20	29	2003-02-08	2003-02-10	2003-02-10	2003-02-12	5	Good Sup...	Justin Glory	(415) 665...
23	30	2003-02-08	2003-02-10	2003-02-10	2003-02-12	5	Good Sup...	Justin Glory	(415) 665...
25	31	2003-02-08	2003-02-10	2003-02-10	2003-02-12	5	Good Sup...	Justin Glory	(415) 665...
7	32	2003-02-12	2003-02-14	2003-02-16	2003-02-15	10	American ...	Sebastian...	(610) 939...
12	33	2003-02-13	2003-02-15	2003-02-16	2003-02-15	9	Happy Su...	Rachel M...	(650) 343...

10. Close the ViewSupplier tab.

Building and Executing the Sales Information View

To obtain product sales information, you use the ds_orders data source and the transformation productCatelog_xform.

- The orders data source contains order details in the orderdetails table.
- The transformation productCatelog_xform contains product catalog information.

To build and execute the sales information view

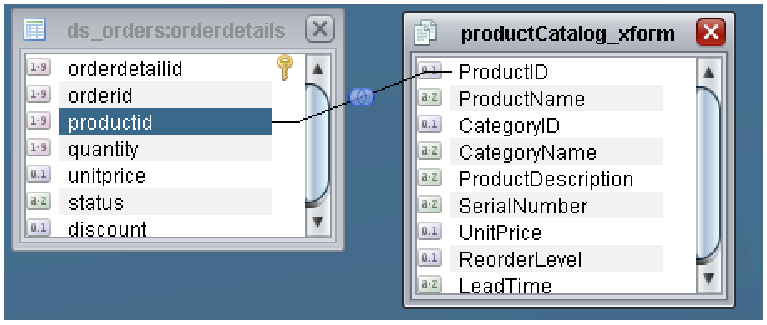
1. Right-click Shared > sources in the resource tree.

2. Create a new view named ViewSales.

3. From the ds_orders data source, drag the orderdetails table into the Model panel.

4. Drag the productCatelog_xform into the Model panel.
5. Click OK on the empty window that pops up.
6. Join **productid** in orderdetails with **ProductID** in productCatelog_xform.

   The join between two different types of resources, one derived from a relational table, and the other from an XML data source.



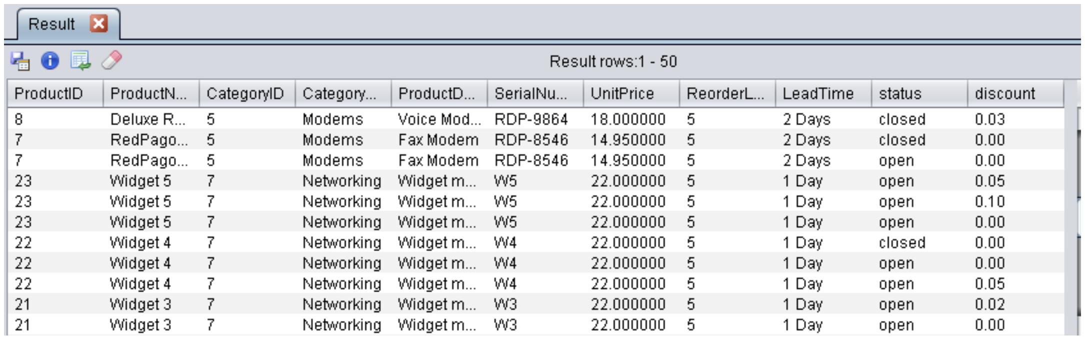
7. In the Grid panel, limit the query to the following columns:

```
productCatelog_xform.*
orderdetails.status
orderdetails.discount
```

8. Save the view.
9. Execute the view.
10. In the result set displayed in the Result panel, locate one of the three rows where ProductID = 23 with the following data:

```
ProductID: 23
```

   These results contain information about the product name, the lead time for product delivery, and other data relevant for the current example.



ProductID	ProductN...	CategoryID	Category...	ProductD...	SerialNu...	UnitPrice	ReorderL...	LeadTime	status	discount
8	Deluxe R...	5	Modems	Voice Mod...	RDP-9864	18.000000	5	2 Days	closed	0.03
7	RedPago...	5	Modems	Fax Modem	RDP-8546	14.950000	5	2 Days	closed	0.00
7	RedPago...	5	Modems	Fax Modem	RDP-8546	14.950000	5	2 Days	open	0.00
23	Widget 5	7	Networking	Widget m...	W5	22.000000	5	1 Day	open	0.05
23	Widget 5	7	Networking	Widget m...	W5	22.000000	5	1 Day	open	0.10
23	Widget 5	7	Networking	Widget m...	W5	22.000000	5	1 Day	open	0.00
22	Widget 4	7	Networking	Widget m...	W4	22.000000	5	1 Day	closed	0.00
22	Widget 4	7	Networking	Widget m...	W4	22.000000	5	1 Day	open	0.00
22	Widget 4	7	Networking	Widget m...	W4	22.000000	5	1 Day	open	0.05
21	Widget 3	7	Networking	Widget m...	W3	22.000000	5	1 Day	open	0.02
21	Widget 3	7	Networking	Widget m...	W3	22.000000	5	1 Day	open	0.00

11. Close the ViewSales tab.

   After adding the data sources to the TDV metadata environment, you created three views as follows:

   – View 1 to obtain order information for the Order department.

   By executing this view, you were able to view the activities of the order department at ALPHA. You learned about the product status, order date, and customer contact information.

- View 2 to obtain supplier information for the Purchase department.

  By executing this view, you were able to learn about ALPHA’s purchase department’s interactions with the customer and the supplier.

- View 3 to obtain sales information for the Sales department.

  By executing this view, you were able to use the sales department’s record and noted the lead time for product delivery.

You will query these views and create a single unified view representing the solution for the customer-reported issue at ALPHA.

Creating a Composite View

For the Sales department at ALPHA, you can build a single, composite view of ALPHA’s business data from the three individual views you have created: ViewOrder, ViewSupplier, and ViewSales. There is no need to access the data sources any more.

To create the composite view

1. Expand Shared > sources in the resource tree.
2. Create a new view named CompositeView.
3. Drag the following views and drop them into the Model panel of CompositeView:
   - ViewOrder
   - ViewSales
   - ViewSupplier
4. Join ProductID in ViewOrder with ProductID in ViewSales.
5. Join ProductID in ViewSales with ProductID in ViewSupplier.
6. In the Grid panel, limit the query to the following columns:

   ```
   ViewOrder.*
   ViewSales.ProductName
   ViewSales.LeadTime
   ViewSupplier.TransactionID
   ViewSupplier.DateRequired
   ViewSupplier.DatePromised
   ViewSupplier.ShipDate
   ViewSupplier.SupplierID
   ViewSupplier.SupplierName
   ViewSupplier.SupplierContactName
   ViewSupplier.SupplierPhoneNumber
   ```

7. Save the view.
8. Execute the view. Result rows 1-50 are displayed.
9. In the result set displayed in the Result panel, look for a row where OrderID = 24 and Transaction ID = 30. There are multiple rows that fit this criteria in the result rows 1-150.
10. Click Load More Results in the Result panel to browse more rows.

    The rows have the following data:

    ```
    OrderID: 24
    ```

```
Status: open
OrderDate: 2003-02-06
CompanyName: Landmark Systems
CustomerContactFirstName: Joyce
CustomerContactLastName: Landers
CustomerContactPhone: (212) 333-1000
ProductID: 23
Discount: 0.05
ProductName: Widget 5
LeadTime: 1 Day
TransactionID: 30
DateRequired: 2003-02-10
DatePromised: 2003-02-10
ShipDate: 2003-02-12
SupplierID: 5
SupplierName: Good Supplies International
SupplierContactName: Justin Glory
SupplierPhoneNumber: (415) 665-8000
```

This result set provides a consolidated view of the customer-reported issue, and also contains information that can be used to find a solution to the current problem. You can contact the customer and offer a better discount. You can also contact the supplier to report the seriousness of the problem and negotiate a price reduction.

CompositeView is the single unified view that represents your business solution.

11. Close the CompositeView tab.

Optionally, you can publish each view to make it available to client applications that connect to the server through JDBC and/or ODBC. To do so, you must publish a view as a TDV database table. After you publish a view, you can reuse it as a relational database table to query further.

For further details on publishing, see the *TDV User Guide*.

Publishing Your Views

The location to publish a view for JDBC or ODBC client applications is Data Services in your Desktop.

TDV data services are the entry points for external applications to communicate with TDV Server and the metadata. Therefore, this is where you will publish the resources that you want to make available for client applications. You will publish the views you have created so far to a TDV data service of the type TDV database, which is similar to other relational databases and which you can query just as you would query a normal database.

This section describes how to create a TDV data service of the type TDV database. For a quick tour of how to create and publish a REST service, see the tutorials in the *TDV User Guide*.

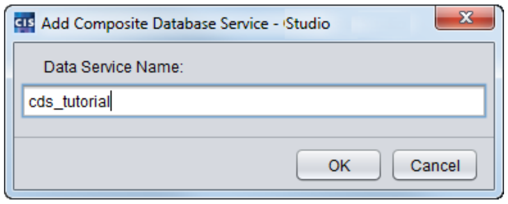
To create a TDV data service of the type TDV database

1. Right-click Data Services > Databases in the resource tree.

2. Select New Composite Database Service.

   The Add Data Service window prompts you to enter a name for the service.

3. Type cds_tutorial, as the unique name for the TDV data service you are creating.



4. Click OK.

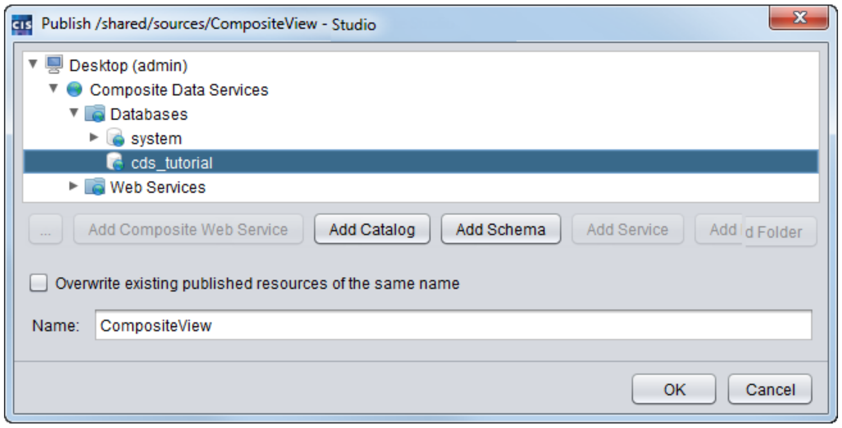
   This entry is similar to a database name, and will display in the resource tree.

Publishing Views to TDV Databases

This section describes how to publish your views to the newly created TDV database (cds_tutorial). The process is similar for publishing other resources.

To publish CompositeView to a TDV database

1. Expand Shared > sources in the resource tree.
2. Right-click CompositeView, and select the Publish option.
3. In the Publish window, specify the location to publish the view, by navigating to Data Services > Databases > cds_tutorial.



   Optionally, you can add a catalog and a schema and specify the schema as the location to publish the view.

4. You can accept the default name displayed in the Name field, and click OK.

   The view is now published.

5. You can expand Data Services > Databases > cds_tutorial in the resource tree to see the view published as CompositeView.
6. Optionally, you can publish the other views.

What You Learned in this Tutorial

During this tutorial:

- You created three separate views.

- You learned how to introspect the underlying physical data sources for selecting only the tables you needed for your modeling, and also to construct views.
- You created views to represent the activities of a particular department (Order, Purchase, or Sales) at ALPHA Manufacturing. Each department conducts its own business without having to share information with another department so the views represent internal organizational activities.
- You modeled the final, unified view from three individual views.
- You learned how to use a view as a building block in its own capacity. Because you can use a TDV database as any relational database, you can build any number or type of views you want until a view matches your business need and publish that view as a TDV database table. This task demonstrates the reusability of a view.
- You learned how to publish the composite view (CompositeView). The view was based on three separate views (ViewOrder, ViewSupplier and ViewSales). Published views are like virtual database tables and you can query them as you would the tables in a relational database.

There are many more modeling and publishing tasks you can perform using TDV, such as transforming data, creating procedures, publishing views and procedures as TDV databases and Web services, and setting up security.

For details on security and other modeling tasks, see the *TDV User Guide*.

Sample Resources

This topic gives an overview of the sample resources that are available when you launch Studio.

Two relational data sources are used in this example. The inventory data source stores information about ALPHA's inventory transactions, products, purchase orders, and suppliers. The orders data source stores information about ALPHA's customers, order details, orders, and shipping methods.

The third data source is an XML product catalog. It has information about ALPHA's products including prices, reorder level, and lead time.

Topics covered in this topic include:

- Sample Data Sources, page 37
- Sample Transformations, page 37
- Sample Definition Set, page 38
- Sample Published View, page 38
- Sample Views, page 38
- Sample SQL Script, page 38
- Sample Discovery Model, page 38

Sample Data Sources

These physical data sources preloaded with data are in the TDV metadata repository.

Logical Names	Resource Tree Name	Data Source Type	Table Names
orders	ds_orders	Set of relational tables	inventorytransactions products purchaseorders suppliers
inventory	ds_inventory	Set of relational table	customers orderdetails orders shippingmethods
productCatalog.xml	ds_XML	XML file	productCatalog.xml

Sample Transformations

The sample transformations are displayed in the resource tree:

- getInventoryTransactions—Transforms tabular data from different sources into XML using the InventoryTransactions definition set.
- productCatelog_xform—Transforms the XML data in `ds_XML` into tabular form.

- productCatelog_transformation–A legacy transform procedure that we want to convert to the updated transform format.

Sample Definition Set

A sample XML-type definition set, InventoryTransactions, is provided with TDV. You can use it to create other resources such as a transformation similar to getInventoryTransactions.

Sample Published View

Resources in Desktop > Data Services are considered published.One sample view, CompositeView, was created during the tutorial steps.

Sample Views

The following sample views exist in the resource tree:

View Name	Description
CompositeView	This view provides a unified, composite view of other views, ViewOrder, ViewSales, and ViewSupplier. It examines the three separate views and joins them on the ProductID column.
ViewOrder	This view provides a specific view of the ds_orders data source and retrieves details about orders and customers. It examines three tables from the ds_orders data source, and joins the tables on two columns, CustomerID and OrderID. Then, it selects specific columns for the result set.
ViewSales	This view provides a combined view of a relational data source (ds_orders) and an XML-type of file data source (productCatalog.xml). The view examines the orderdetails table from ds_inventory and the entire productCatelog_xform, and joins them on one column, ProductID. Then, it selects specific columns for projection in the execution result set.
ViewSupplier	This view provides a specific view of the ds_inventory data source and retrieves information about inventory and suppliers. It examines three tables from the ds_inventory data source, and joins the tables on two columns, PurchaseOrderID and SupplierID. Then, it selects specific columns for the result set.

These views are unpublished and not ready for external client access. For details on views, see the *TDV User Guide*.

Sample SQL Script

A SQL script, LookupProduct, has been precreated. You can use this script in other resources, for example the sample transformation named getInventoryTransactions.

Sample Discovery Model

Examples Data Model can be opened to get an idea of what kind of information you can gather when using Discovery.