TIBCO® Data Virtualization

Getting Started Guide

Version 8.8.1 | March 2025

Contents

Introduction

TIBCO® Data Virtualization (TDV) is data virtualization software that lets you integrate data quickly and cost effectively. With TDV, you can build and manage virtualized views and data services that access, transform, and deliver the data your business requires to accelerate revenue, reduce costs, lessen risk, and improve compliance. TDV is:
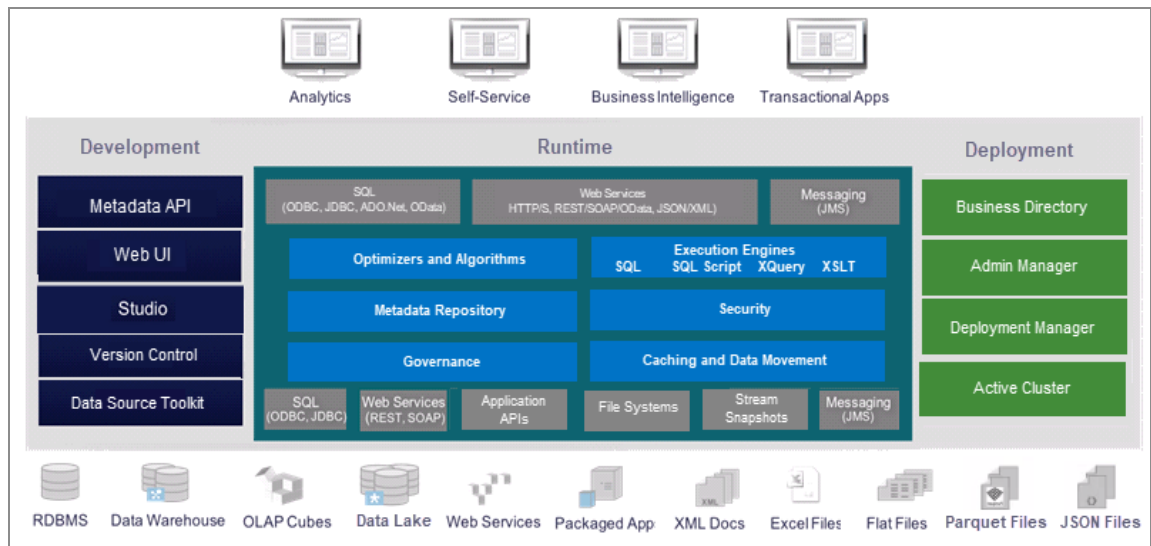
- **Fast and Economical** – Integrate data reliably at a fraction of the time, cost, and rigidity of physical warehousing and ETL. Evolve rapidly when requirements change.

- **Immediate** – Deliver data as needed, using advanced performance optimization algorithms and fine-grained security.

- **Business-friendly** – Transform native IT structures and syntax into easy-to-understand, IT-curated data sets sharable through a self-service business directory.

- **Wide-ranging** – Access data from distributed data sources including traditional enterprise, big data, cloud, and IoT. Use it across myriad analytics, self-service, business intelligence and transactional applications.

- **Enterprise Grade** – Support multiple lines of business, hundreds of projects and thousands of users.

The following topics are discussed in this chapter:

- TIBCO® Data Virtualization Architecture
- About TDV Roles
- TDV Resource Terms
- Studio Modeling and Publishing Terms

TIBCO® Data Virtualization Architecture

TDV is Java-based, enterprise-grade middleware whose modular structure supports all phases of data virtualization development, run-time, and management.

Product	Description	For More Information
Business Directory	Business Directory is a self-service business data directory you can use to easily search, categorize, and consume IT-curated data sets developed using TDV. Business Directory encourages data set sharing and reuse to accelerate business outcomes while reducing IT workloads.	*Business Directory Guide*
Discovery	Discovery enables you to go beyond simple data profiling to examine data, locate important entities and reveal hidden relationships across distinct data sources. You can quickly build and display comprehensive E-R diagrams and data models to meet new business requirements faster.	*Discovery User Guide*
Web UI	The TIBCO® Data Virtualization Web UI is a self service data provisioning web user interface that enables you to create and publish your own views that can then be consumed in your 3rd party downstream apps. The TDV Web UI provides an agile way to transform	WebUI User Guide

Product	Description	For More Information
	and move data between different data sources. These data sources can be located anywhere, even in the cloud. Using the easy to use web interface, the data engineers, TDV developers and business analysts can create datasets, perform complex SQL queries to manipulate data, export and publish the result set. The complex data processing workflows created are reliable, repeatable and secure. The easy to use graphical interface's drag and drop feature lets you do all these with limited or no knowledge of SQL.	
Studio	Studio is the agile modeling, development, and resource management tool for your data. It is oriented for developers to model data, design views and data services, build transformations, optimize queries, and manage resources. Studio's graphical modeling environment provides a flexible workspace where queries are created and tested, and a data services repository where completed objects are published. Studio offers a rich set of transformations and transformation editors. Development languages including SQL, SQL Script, Java, XQuery, and XSLT complement Studio's graphical modeling capabilities.	User Guide
Cost-Based and Rules-Based Optimizers	Cost-Based and Rules-Based Optimizers used by the Federation Engine work directly with data source optimizers to maximize query performance. The Objects Repository lets you manage your data services throughout their lifecycle. Fine-grained Security capabilities, including authentication, authorization, and encryption, safeguard that you deliver sanctioned data only. Flexible Caching options enable higher	User Guide

Product	Description	For More Information
	performance and greater uptime. Quality helps ensure that you deliver the best data possible. Built-in Governance features provide complete visibility, traceability, and control.	
Adapters	Adapters provide a wide range of data source connectivity for databases, files, big data, cloud sources, and packaged applications. Beyond schema-to-schema only connectivity, TDV adapters integrate with data source optimizers to ensure more accurate queries and higher performance. The Data Source Tool Kit allows you to build your own custom adapters.	*Adapter Guide*
Manager	Manager is the administrative console your administrators use to set up user IDs, passwords, security profiles, view logs, and check server activity.	*Administrator Guide*
Deployment Manager	Deployment Manager lets you migrate entire projects in a single step (including their resources, cache settings, server configurations, security profiles and more) across instances to simplify and automate your development lifecycle.	*Administrator Guide*
Monitor	Monitor provides a comprehensive, real-time view of your TDV server environments. Monitor displays all the pertinent system health indicators required to help your IT operations staff guide corrective actions.	*Monitor Guide*
Active Cluster	Active Cluster works with your load balancers to provide high availability and greater scale to meet your challenging service-level-agreements. Active Cluster simplifies complex operations management by automatically sharing resources, adjusting capacity on demand, and more.	*Active Cluster Guide*

About TDV Roles

As a user of the TIBCO® Data Virtualization, you may be one or more of the following:

- End user

- Operator

- IT Developer

- Analyst

- Administrator

Your role decides which TDV component you can use, and what functionality you have access to. The following table lists the different components of TDV and which TDV Persona it is more applicable for:

Component	Function	End User	Operator	IT Developer	Analyst	Administrator
Published Data Services	Access the published data services for consumption by client applications.	Yes			Yes	
Business Directory	Search and govern the enterprise catalog for data.			Yes	Yes	Yes
Web UI	Create data flows using the self service tool (drag and drop) and to view published resources.				Yes	

Component	Function	End User	Operator	IT Developer	Analyst	Administrator
Studio	Create data sources, design and develop TDV resources, build transformations, optimize queries and manage TDV resources.			Yes		Yes
Manager	Observe current system state, manage security, cluster and workload management.	Yes	Yes			Yes
Deployment Manager	Move a solution through the development, test and production stages.					Yes

TDV Resource Terms

TDV resource refers to the resources that are used for data modeling and building business solutions using TDV software. Data stored in these resources are available in tabular or hierarchical format, and noted accordingly as either tabular data or hierarchical data. 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*.

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

Studio Modeling and Publishing Terms

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. Studio 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.

For details, see the *TDV User Guide*.

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
- Studio Modeler and Resource Tree Overview
- Viewing and Opening Resources

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 > TIBCO > TDV <version> > Studio > Start Studio <version>.

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

Field	Tutorial Values	Description
Language	English	Choose the language you want your TDV application in. Note: Any other TDV application (WebUI, Deployment Manager or Web Manager) you access from TDV Studio will be displayed in the language you choose when logging into TDV Studio.
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.

Field	Tutorial Values	Description
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 TDV Server. Otherwise, proceed to sign in using username and password authentication.

If an OAuth 2.0 domain has been set up in Web Manager and access tools privileges given, then you can login to TDV Studio by choosing the domain name from the drop down or by entering the name in the Domain field. You will not need to provide username and password while using the OAuth 2.0 domain.

Notes:

- You need to be actively logged into your IdP account, before attempting to login to any of the TDV client tools using the OAuth2 domain. If not, you will be redirected to your IdP login page. Once logged in, click on the Oauth2 domain in TDV Studio, Web Manager or Web UI to successfully login to the TDV client.

- In order to allow access to Studio, the TDV Studio url must be added to the Redirect URIs list in your Identity Provider. Using the following hostname based url is the preferred method:

      http://xyz.com/oauth2studiocallback

      https://xyz.com/oauth2studiocallback

   You may also add an IP based or localhost based url in the Redirect URI list as given below:

      http://1.1.1.1:9400/oauth2studiocallback

      https://1.1.1.1:9402/oauth2studiocallback

> http://localhost:9400/oauth2studiocallback
>
> https://localhost:9402/oauth2studiocallback

However, it is important to note that this is not a preferred approach as:

– localhost references are not host specific.

– IP addresses will not work in deployment environments where server IP addresses are not statically assigned (For example, in the cloud deployments).

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.

Note: Logging out of TDV Studio invalidates the token that was used to authorize the user. Some providers do not support access token revocation. Refer to your idP documentation for the specific help regarding renewal/revocation of access and refresh tokens.

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.
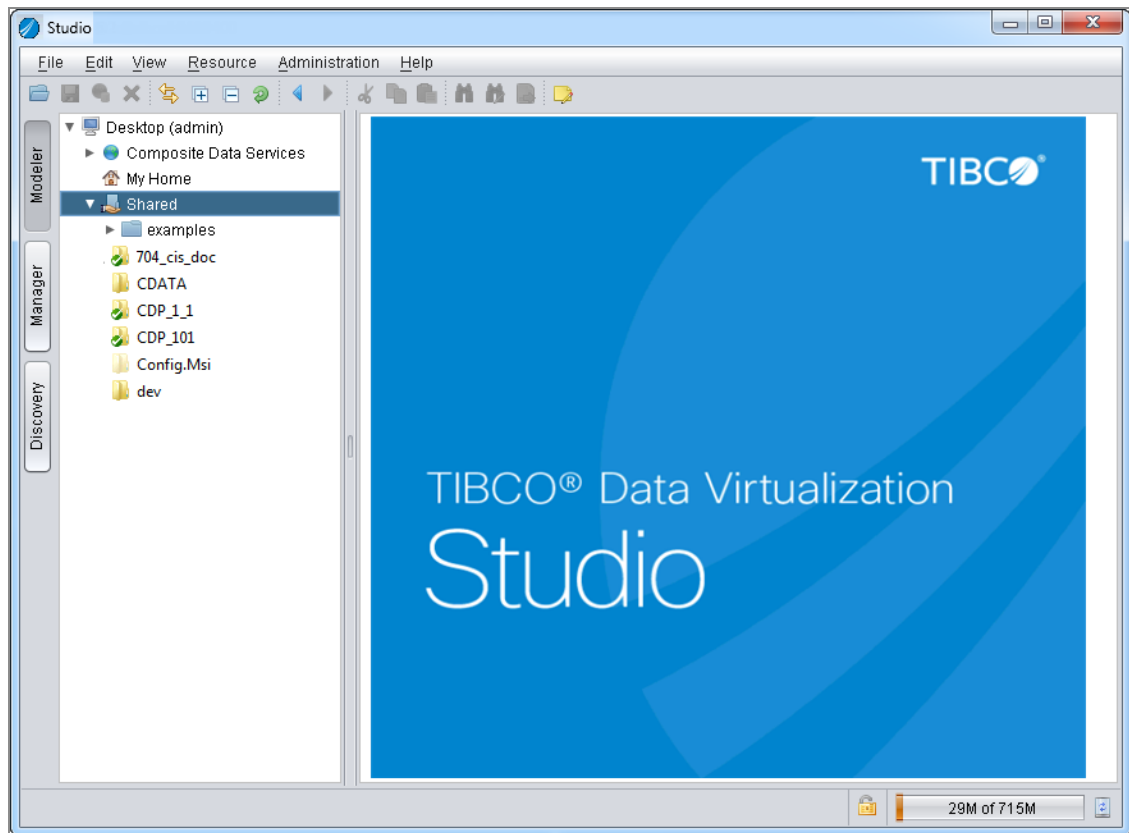
- Desktop—The current user's virtual work area in TDV Server, and is like the desktop for a personal computer's user interface.

- 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 Studio.

To open a resource

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

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

Getting Started with Studio

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

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 Studio. 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
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

Task	Description	Instructions
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
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
6	Publish the unified views as TDV database tables. This unified view provides a single view solution for the business issue.	Publishing Your Views

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
- Adding the Orders Data Source
- Viewing the Data Source Schema
- Adding the Inventory Data Source
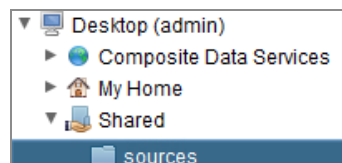
- Adding the XML Data Source

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.

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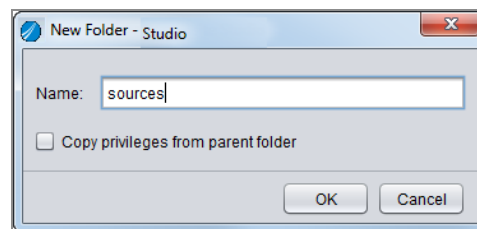
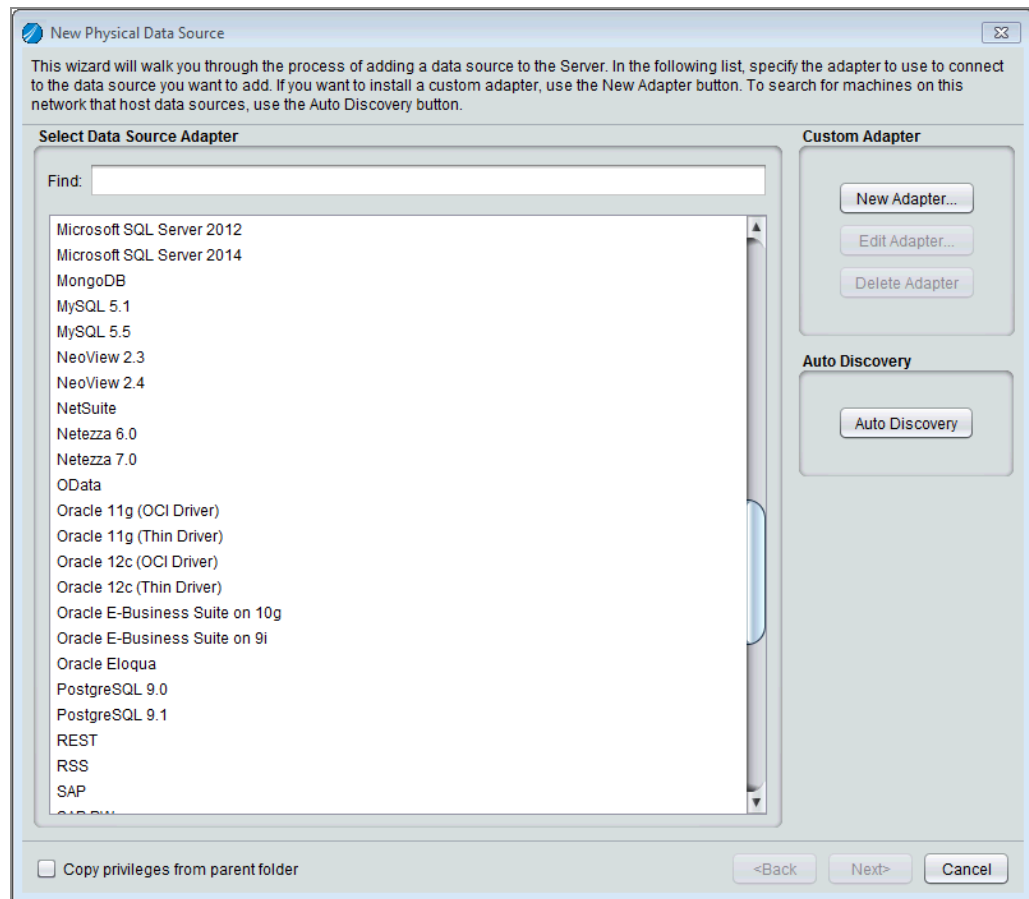The newly created folder is added to the resource tree under Shared.

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.

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.

Field Name	Tutorial Value	Description
		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.
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*.

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.
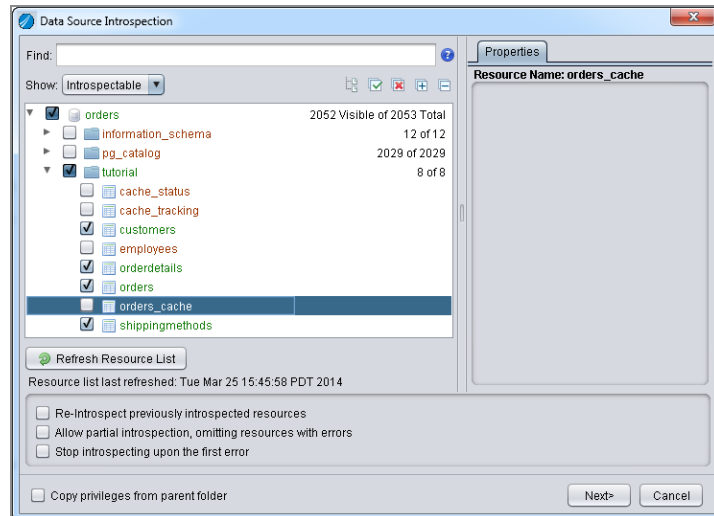
9. Click Next.

10. Click Finish.
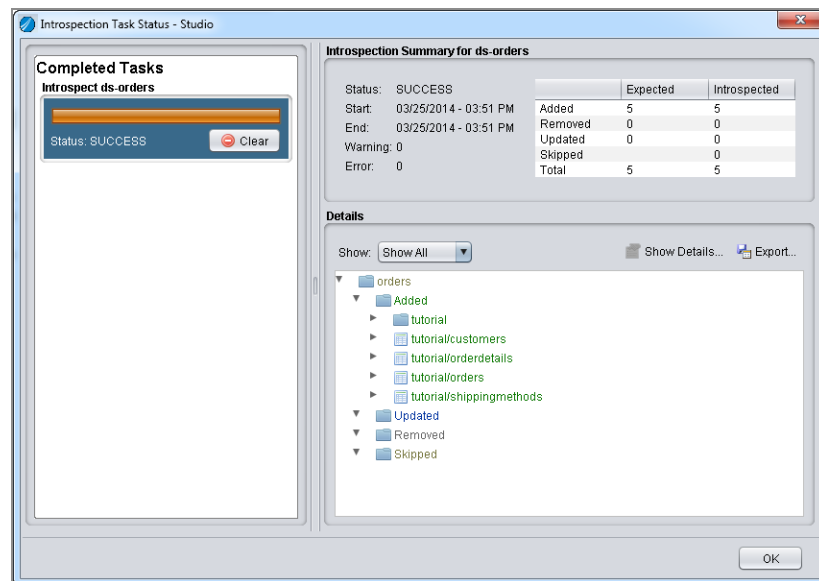
11. Click OK.

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.

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.
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.

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
```
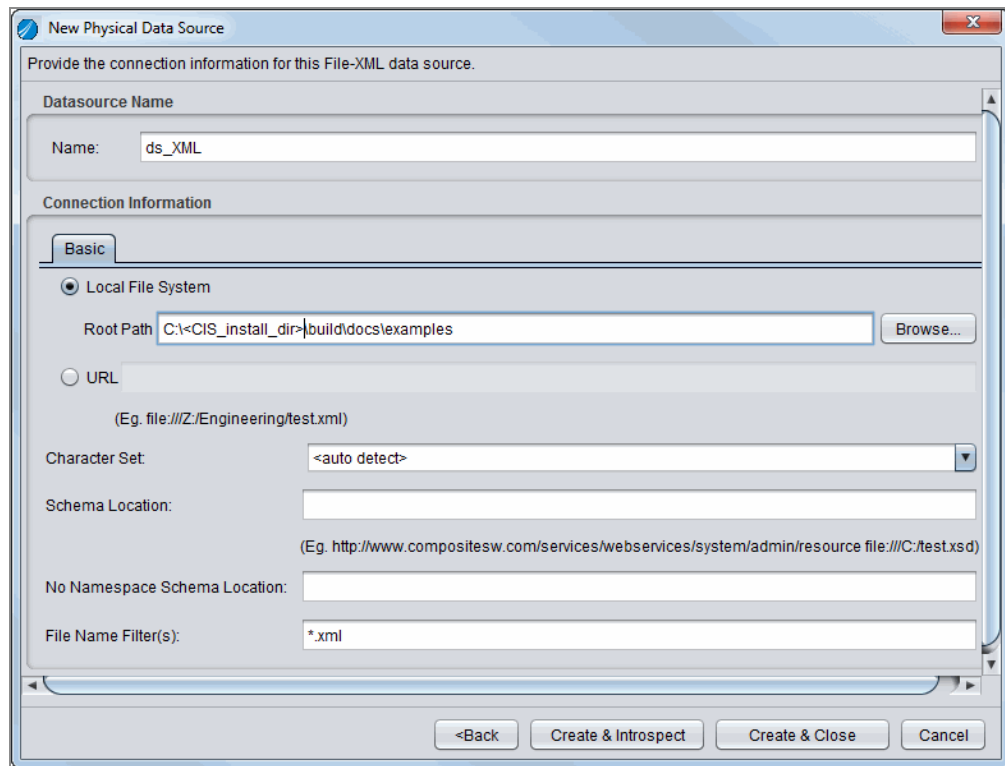
To add the XML data source

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.

Studio 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.

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.
		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)		

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.
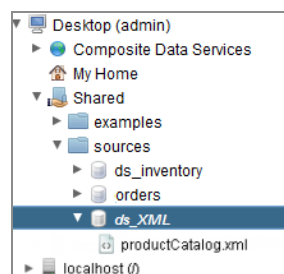
Creating the XML Data Transformation

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
- Adding Loop and Cast Functions

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:
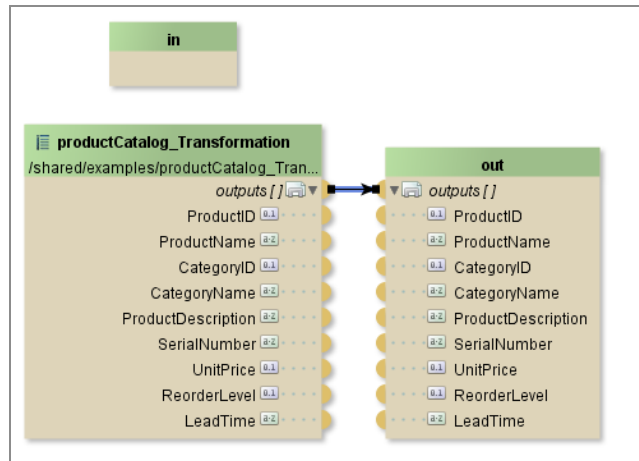
9. Delete productCatelog_transform. The out operation should retain all the necessary outputs and their data types.

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:
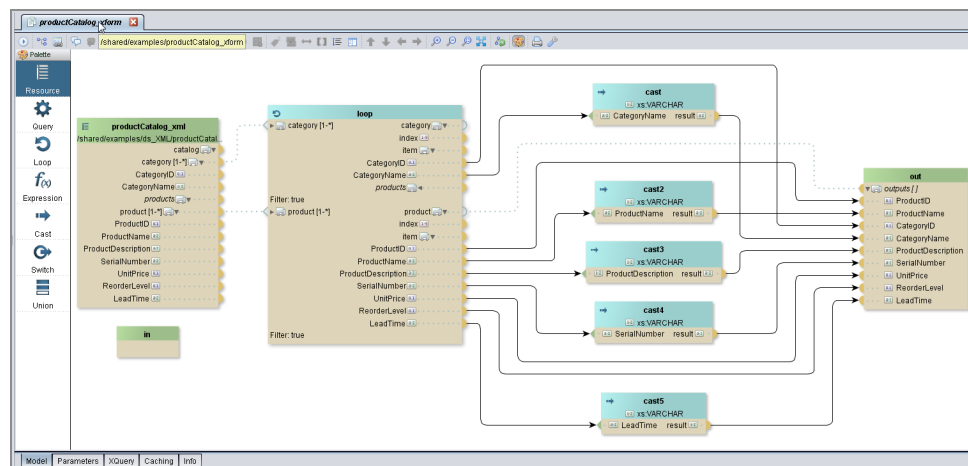
To add loop and cast operations to your transform

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:
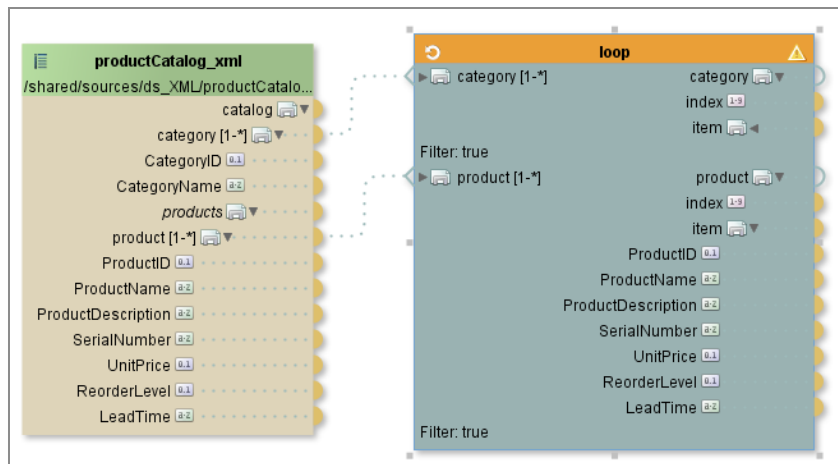
6. Connect products in the loop operation to outputs in the out operation.

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.

Loop Element	Out Operation Name	Insert Cast
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
- Building and Executing the Supplier Information View
- Building and Executing the Sales Information View

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.
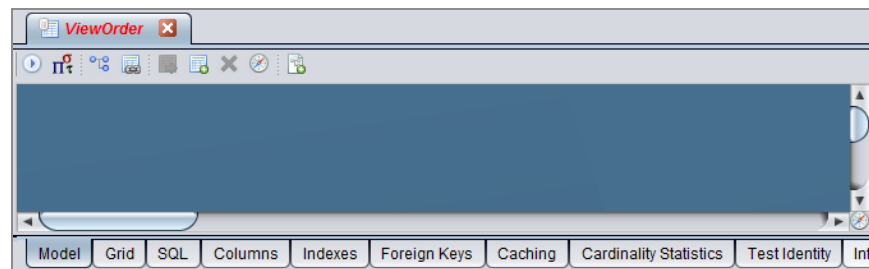
   The editor has the following tabs:

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

Tab	Description
	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.
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.
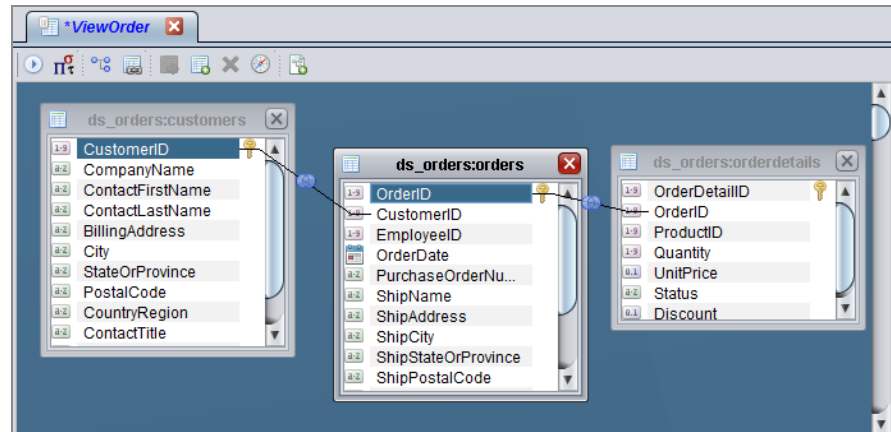
For details on joins, see the *TDV User Guide.*

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:

   Click the first cell under Column.

   Select orderdetails.orderid in the drop-down list.

   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.

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.

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:

   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
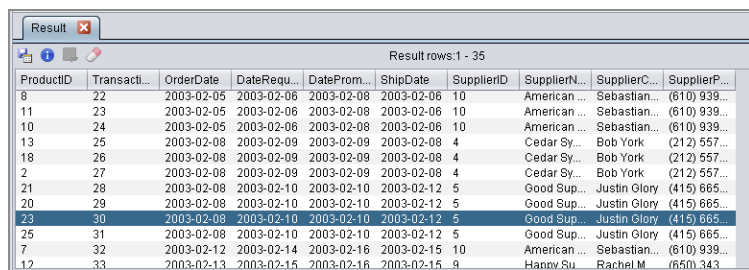
SupplierName: Good Supplies International

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).

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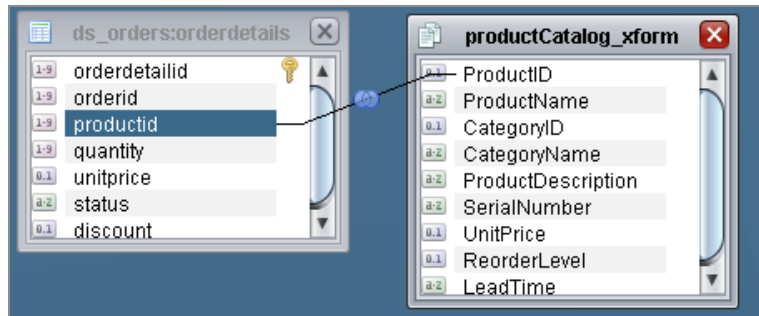
7. In the Grid panel, limit the query to the following columns:

    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.

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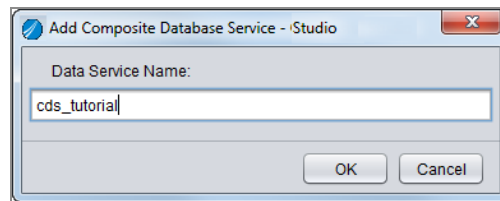
This section describes how to create a TDV data service of the type TDV database. For a quick tour of how to create and publish a REST service, see the tutorials in the *TDV User Guide*.

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.
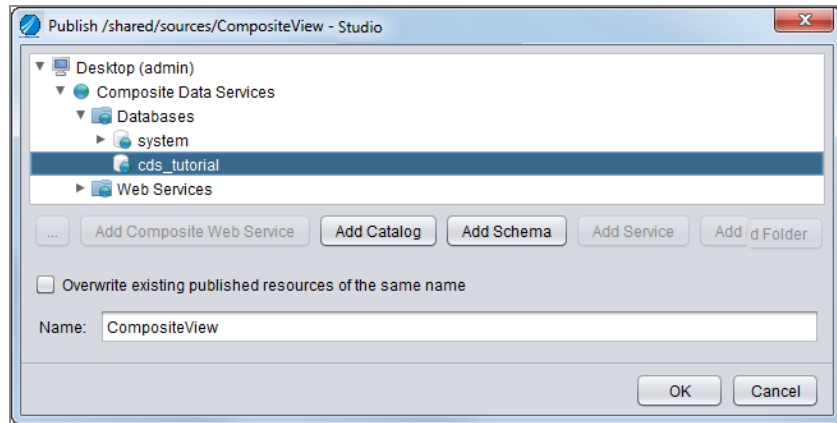

4. Click OK.

   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.

- Sample Data Sources
- Sample Transformations
- Sample Definition Set
- Sample Published View
- Sample Views
- Sample SQL Script
- Sample Discovery Model

Sample Data Sources

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

	Resource Tree Name	Data Source Type	Table Names
orders	ds_orders	Set of relational tables	inventorytransactions products purchaseorders suppliers

	Resource Tree Name	Data Source Type	Table Names
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.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the TIBCO® Data Virtualization page.

Users

      TDV Getting Started Guide

      TDV User Guide

      TDV Web UI User Guide

      TDV Client Interfaces Guide

      TDV Tutorial Guide

      TDV Northbay Example

Administration

      TDV Installation and Upgrade Guide

      TDV Administration Guide

      TDV Active Cluster Guide

      TDV Security Features Guide

Data Sources

      TDV Adapter Guides

TDV Data Source Toolkit Guide (Formerly Extensibility Guide)

References

TDV Reference Guide

TDV Application Programming Interface Guide

Other

TDV Business Directory Guide

TDV Discovery Guide

TDV and Business Directory Release Notes - Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Release Version Support

TDV 8.5 and 8.8 are designated as Long Term Support (LTS) versions. Some release versions of TIBCO® Data Virtualization products are selected to be long-term support (LTS) versions. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also Long Term Support.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.

- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking Register on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote

on feature requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. ("CLOUD SG") SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, "INCLUDED SOFTWARE"). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, TIBCO logo, TIBCO O logo, ActiveSpaces, Enterprise Messaging Service, Spotfire, TERR, S-PLUS, and S+ are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG's Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL