IBM Cognos Business Intelligence Virtual View Manager

Version 10.1.0

Getting Started Guide
# Table of Contents

**Introduction**  
5

**Chapter 1: Overview of IBM Cognos Virtual View Manager**  
7  
- Software Components  
- Resources  
  - Data Sources  
  - Views  
  - Procedures  
  - Definition Sets  
  - Triggers  
- IBM Cognos Virtual View Manager Data Services  
- Databases  
- Web Services  
- Folders  
- Tabular and Hierarchical Data  
- Modeling and Publishing  
  - Introspecting  
  - Creating  
  - Publishing  

**Chapter 2: Logging Into IBM Cognos Virtual View Manager**  
13  
- Starting IBM Cognos Virtual View Manager and Connecting to the Server  
13

**Chapter 3: Sample Resources**  
15  
- Starting IBM Cognos Virtual View Manager  
- Example Resources  
- Sample Data Sources  
- Sample Transformations  
- Sample Views  
- Sample Definition Set  
- Sample SQL Script  
- Sample Published View  

**Chapter 4: Getting Started with IBM Cognos Virtual View Manager**  
21  
- About this Tutorial  
- Tutorial Scenario  
  - Data Sources Where Information Is Stored  
- Tutorial Work Flow  
- Adding Data Sources  
  - Summary  
  - What’s Next After Adding the Data Sources?  
- Building Simple Views  
  - View 1: Obtaining Order Information  
  - View 2: Getting Supplier Information  
  - View 3: Obtaining Sales Information  

Licensed Materials – Property of IBM  
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>37</td>
</tr>
<tr>
<td>Creating a View</td>
<td>37</td>
</tr>
<tr>
<td>What’s Next After Building the Federated View?</td>
<td>39</td>
</tr>
<tr>
<td>Publishing Your Views</td>
<td>39</td>
</tr>
<tr>
<td>Publishing Views to IBM Cognos Virtual View Manager Databases</td>
<td>40</td>
</tr>
<tr>
<td>Summary</td>
<td>40</td>
</tr>
<tr>
<td>Salient Features</td>
<td>40</td>
</tr>
<tr>
<td>What’s Next?</td>
<td>41</td>
</tr>
</tbody>
</table>

Index 43
Introduction

The purpose of this guide is:

- To demonstrate how you can use IBM® Cognos® Virtual View Manager to address your business needs
- To introduce the data modeling aspect of Virtual View Manager

Audience

This documentation is for information technology professionals who want to use IBM® Cognos® Virtual View Manager to model data resources. Knowledge of relational data sources, hierarchical data sources, and data modeling is recommended.

This guide is designed for first-time users that are not administrators and are interested in addressing business issues presented by their disparate business systems.

Finding information

To find IBM® Cognos® product documentation on the web, including all translated documentation, access one of the IBM Cognos Information Centers at http://publib.boulder.ibm.com/infocenter/cogic/v1r0m0/index.jsp. Updates to Release Notes are published directly to Information Centers. You can also read PDF versions of the product release notes and installation guides directly from IBM Cognos product disks.

Accessibility features

This product does not currently support accessibility features that help users who have a physical disability, such as restricted mobility or limited vision, to use this product.

Forward-looking statements

This documentation describes the current functionality of the product. References to items that are not currently available may be included. No implication of any future availability should be inferred. Any such references are not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of features or functionality remain at the sole discretion of IBM.

Samples disclaimer

The Great Outdoors Company, GO Sales, any variation of the Great Outdoors name, and Planning Sample depict fictitious business operations with sample data used to develop sample applications for IBM and IBM customers. These fictitious records include sample data for sales transactions, product distribution, finance, and human resources. Any resemblance to actual names, addresses, contact numbers, or transaction values is coincidental. Other sample files may contain fictional data manually or machine generated, factual data compiled from academic or public sources, or data used with permission of the copyright holder, for use as sample data to develop sample appli-
Introduction

cations. Product names referenced may be the trademarks of their respective owners. Unauthorized duplication is prohibited.
Chapter 1: Overview of IBM Cognos Virtual View Manager

This chapter provides a high-level overview of the concepts underlying IBM® Cognos® Virtual View Manager, and introduces the resources that are available to use the software and benefit from its capabilities.

Virtual View Manager offers a virtual data management system for Enterprise Information Integration (EII). It provides quantitative business analysts, application/report developers, and the applications they build a secure and re-usable unified view solution that requires no change to existing end-user applications or underlying information infrastructure.

Virtual View Manager is comprised of a server, a client application, and JDBC/ODBC drivers for the enterprise client programs to communicate with the server via JDBC/ODBC.

Virtual View Manager manages the complexity of interacting with disparate, distributed information sources, creating a transparent, real-time interface to business information for business users and application developers. It creates a securely managed unified view across files, databases, and packaged applications.

Virtual View Manager supports a wide range of data sources such as Oracle, Microsoft® SQL Server, MySQL, IBM® DB2®, Sybase, IBM Informix®, Netezza, Microsoft Access, Microsoft Excel, LDAP, flat files (including data in XML), and Web services. The server connects to these data sources through JDBC, ODBC, SOAP, and so on.

From a client application of your choice, using Virtual View Manager, you can browse, query, update, and manage information across the enterprise and any number or type of data sources.

Software Components

IBM® Cognos® Virtual View Manager is comprised of the following components:

- the Server represents the core runtime environment that hosts various components such as the query engine and metadata repository
- the Virtual View Manager studio application provides an interface for data modeling, querying, transformation, and administration
- the JDBC driver lets client programs access the server through JDBC
- the ODBC driver lets client programs access the server through ODBC

Resources

This document uses the term resources to collectively refer to the resources that are used for data modeling and building business solutions. These resources are data sources, views, parameterized queries, SQL scripts, Java™ procedures, packaged queries, transformations, and IBM® Cognos®
Virtual View Manager data services (which are available as Virtual View Manager databases and Web services). Data stored in these resources are available in tabular or hierarchical format, and noted accordingly as either tabular data and hierarchical data.

Resource names are the parent container path combined with the resource name to provide a unique identifier for the invocation and reference to any defined resource.

The next several paragraphs describe Virtual View Manager resources, tabular data, and hierarchical data.

**Data Sources**

A data source is the representation of the external, underlying physical data source that is available to the server.

The data sources form the foundation layer for building queries in the Virtual View Manager metadata environment. Ultimately, it is the data found within the tables, XML documents, LDAP hierarchies, and procedures of these data sources that you want to see combined across information sources into a unified view to address your business questions, reports, or processes. In Virtual View Manager, this combined view is referred to as view.

Refer to the chapter *Data Sources* in the IBM® Cognos® Virtual View Manager User Guide for more information.

**Views**

A view is a resource through which you can access and integrate data from any number of underlying physical data sources.

In its simplest form, a view is an SQL SELECT statement against one or more data sources. Views are data retrieval statements that can be executed directly from the Modeler or published as Virtual View Manager database tables and Web services. See also "IBM Cognos Virtual View Manager Data Services" (p. 10).

For further details on views, refer to the chapter *Views* in the User Guide.

**Procedures**

Procedures are programs that are used to generate or act on data, much like a SELECT or an UPDATE statement. Virtual View Manager supports several types of procedures, which are described next.

**SQL Scripts**

SQL scripts are scripts written in Virtual View Manager’s own SQL Script language. SQL Script is similar to other scripting languages used by relational databases for implementing stored procedures, and serves the same purpose. The mechanism is to allow logic to be performed in the server.

For further details on SQL scripts, refer to the chapters *Procedures* in the User Guide and *Virtual View Manager SQL Script* in the Reference Manual.
Custom Java Procedures

Custom Java™ procedures are programs that you write in Java to run inside the Virtual View Manager server for your specific use. Virtual View Manager provides the necessary application programming interfaces (APIs) which you need to use in a Java procedure to interact with the Virtual View Manager server.

For further details on custom Java procedures, refer to the chapters Procedures in the IBM® Cognos® Virtual View Manager User Guide and Java APIs for Custom Procedures in the IBM® Cognos® Virtual View Manager Reference Guide.

Packaged Queries

A packaged query is a type of query you can create in Virtual View Manager to use database-specific queries within the Virtual View Manager system. Sometimes, you may already have a complex query written for a particular database and it may not be feasible or desirable to re-write that query in Virtual View Manager’s SQL. In such cases, the database-specific query can be employed as a packaged query and subsequently be used in other queries.

For further details on packaged queries, refer to the chapter Procedures in the IBM® Cognos® Virtual View Manager User Guide.

Parameterized Queries

A parameterized query is essentially an SQL SELECT statement that can have named parameters in the projections or selections, and it is implemented as a single-statement SQL script.

For further details on parametrized queries, refer to the chapter Procedures in the User Guide.

Transformations

In Virtual View Manager, you query data stored in tabular format as rows and columns. If you want to query data stored in a hierarchical format, as in XML, then you should first convert that data to fit into a tabular format. This conversion process is called transformation. Transformation also refers to the transformed data.

When hierarchical data is flattened into a table or set of tables, you can combine them with relational database tables to build views.

For further details on transformations, refer to the chapter Procedures in the User Guide.

Definition Sets

A definition set is a library of user-defined XML schema or SQL definitions. You can refer to the definitions in a definition set from any procedure and also share them with other users in the system. Any change you make in a definition is automatically propagated through all the procedures that use the definition.

For further details on definition sets, refer to the chapter Definition Sets in the User Guide.

Triggers

Trigger is a resource that triggers a specific system activity to be performed under specified conditions while such activity is not started as a result of any user interaction.
For specific details on triggers, see the chapter *Triggers* in the *User Guide*.

**IBM Cognos Virtual View Manager Data Services**

IBM® Cognos® Virtual View Manager Data Services provide a public interface so that external clients may gain access to data services, resources, and operations published by Virtual View Manager.

Resources introspected, configured, and defined by the Virtual View Manager server metadata may be selectively published on the Virtual View Manager Data Services node as either databases or Web services.

Virtual View Manager Data Services are secured by several layers of security:

- Explicit and implicit definition of privileges to select and execute SQL on resources and container directories. Refer to section on *Privileges* in the *User Guide* Chapter on *Security*.

- Virtual View Manager mediated authentication or LDAP mediated authentication. Refer to Chapters 3 and 4 in the *Administration Guide* for more information on Virtual View Manager Domain Administration and LDAP Domain Administration.

- HTTP using SSL/TLS. Refer to "Transport Layer Security (TLS)" and "To secure a Web service port" in the *User Guide*.


Each of these security layers may be applied according to customer requirements.

Publish selected tabular data and procedures to this area so that they are available to JDBC, ODBC, and SOAP client programs. JDBC/ODBC/SOAP client programs are applications/programs that use JDBC/ODBC/SOAP to communicate with Virtual View Manager. With the right set of access privileges, client applications can use Virtual View Manager data services.

For details on how Virtual View Manager data services are created and resources are published, see the chapter *Publishing Resources* in the IBM® Cognos® Virtual View Manager *User Guide*.

Virtual View Manager data services are of two types — Virtual View Manager Databases and Web Services. These publishing types are described next.

**Databases**

Databases are Virtual View Manager databases which are similar to SQL-based relational databases, and they can be queried like traditional databases. In essence, a Virtual View Manager database represents a virtual database in the server, and it models your distributed data the way you want to see it.

In Virtual View Manager, you can publish relational data and procedures as Virtual View Manager database tables.

For further details, see the chapter *Publishing Resources* in the IBM® Cognos® Virtual View Manager *User Guide*. 
Web Services

Web services provide a public interface for publishing data services by SOAP over HTTP, HTTPS, WSIF JMS, or TIBCO JMS. The WSDL and the bindings are defined and described using XML. Any web client can discover and consume data services. These definitions can be discovered by other software systems. These systems may then interact with a web service in a manner prescribed by its definition, using XML-based messages conveyed by Internet protocols.

Supported WS-Security standards


Username Token profile V1.0 (see: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf)

X.509 Token Profile V1.0 (see: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf)

SOAP and Web Services

Simple Object Access Protocol (SOAP) is the formal set of conventions governing the format and processing rules of an XML message. IBM® Cognos® Virtual View Manager supports published data services by SOAP over HTTP, SOAP over HTTPS, SOAP over WSIF JMS, and SOAP over TIBCO JMS.

The Virtual View Manager server automatically generates a WSDL (Web Services Description Language) for resources being published and easily consumes WSDL based web services conforming to the SOAP specification.

The Virtual View Manager server may be used to publish both relational data and procedures as web services.

For more actionable information refer to the chapter Publishing Resources in the User Guide.

Folders

A folder is a convenient resource (or, container) that you can use to store other resources. Folders help you store your resources in an organized manner. For example, you can create a folder to only contain a certain type of resource or resources that can only be used by specific users in the system.

For further details on folders/containers, see the User Guide.

Tabular and Hierarchical Data

The data available for data modeling and querying are normally stored in two forms in the data sources: tabular and hierarchical. The term tabular data refers to the data stored in relational data sources, LDAP directories, and delimited files. The term hierarchical data refers to the data stored in XML and WSDL formats.

Modeling and Publishing

Metadata modeling and publishing in IBM® Cognos® Virtual View Manager is a three-step process described in the following sections.
To facilitate your metadata modeling efforts, you use the Modeler. For details on the Modeler, see the IBM® Cognos® Virtual View Manager User Guide.

**Introspecting**

Introspecting means probing a physical data source and selecting only the desired resources from that data source for the sake of modeling in the Virtual View Manager system. Introspection is a part of connecting to the data source.

Virtual View Manager lets you specify the physical data source you want to probe. When you probe 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. For details on how to connect to the desired physical data source and introspect it, see the chapter Data Sources in the IBM® Cognos® Virtual View Manager User Guide.

**Creating**

Creating means creating, designing, and editing 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 desired performance profile, and subsequently store the view or procedure to be scheduled for batch reporting or to publish as a Virtual View Manager database or Web service.

For details on how to create resources, refer to the IBM® Cognos® Virtual View Manager User Guide.

**Publishing**

Publishing means making the desired views and procedures available to client applications that use JDBC/ODBC/SOAP in the enterprise.

For details on how to publish resources, see the chapter Publishing Resources in the User Guide.
Chapter 2: Logging Into IBM Cognos Virtual View Manager

This chapter describes how to set up the work environment for using the software after installation. The installation process installs the server and other selected components in a specified location and starts its repository database (if the repository has been bundled with the installation).

IBM® Cognos® Virtual View Manager and the Virtual View Manager server are available in a program group on the Start menu. The server starts automatically after installation. So, you only need to launch Virtual View Manager, which is described next.

Starting IBM Cognos Virtual View Manager and Connecting to the Server

To successfully log into the IBM® Cognos® Virtual View Manager Server, you should:

- Obtain a valid user name and password from your administrator
- Know to which domain (cognos or LDAP) you belong
- Know the name of the machine or the IP address where the Virtual View Manager server is installed

For initial evaluation and use, the following login information may be used:

- Username: admin (default value)
- Password: admin
- Domain: cognos (default value)
- Server: localhost (default value)
- Port: 9400 (default value)

Steps

1. From the Start menu, click All Programs, IBM Cognos 8 Virtual View Manager > Studio > Studio.
2. Type the user name in the Username field.
3. Type the password associated with your user name in the Password field.
4. Select your domain from the Domain drop-down list.
   This entry is used for authentication to connect to the server. The default is cognos.
5. Specify the host machine for the server in the Server drop-down list.
This entry defaults to localhost which is the computer where Virtual View Manager running. You can accept this default if Virtual View Manager and the server are running on the same machine. If the server is not running on localhost, you can choose to enter the IP address or the host name of the machine where the server is installed.

6. In the Port field, specify the HTTP base port number through which you can connect to the server.

   The default HTTP base port is 9400. This may have been changed manually, or it may have been changed automatically if a previous version of the server was present prior to installation.

7. To connect to the server securely using SSL via HTTPS simply check the Encrypt check box and all messages passed between the Virtual View Manager session and the server will be encrypted. Otherwise HTTP clear text is used.

8. Click Connect.

   If the log-in credentials are validated successfully, you will connect to the server, and Virtual View Manager will open for your modeling.

When the server is set to accept HTTPS secure mode communications only, a login attempted without the Encrypt check box results in a refused connection and an error message. Dismiss this error if received, and use the Encrypt check box to log in securely.

Proceed to the next chapter for details on what you will see when you first start Virtual View Manager.
Chapter 3: Sample Resources

This chapter gives a high-level overview of the sample resources that are provided. The next chapter provides a tutorial showing how to create some of these resources from scratch.

Starting IBM Cognos Virtual View Manager

Whenever you start IBM® Cognos® Virtual View Manager, the Modeler 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.

When you start Virtual View Manager for the first time after installation, you see some resources displayed in a tree format, especially the top-level nodes and some sub-nodes—Desktop, Virtual View Manager Data Services, My Home, Shared, examples, and <server host machine>—representing the areas to hold the resources for your modeling.

Expand the node labelled examples that is inside the Shared folder, to show the following:

- **Desktop** — This area represents your (current user’s) virtual work area, and is analogous to the desktop in a personal computer’s user interface.

- **Virtual View Manager Data Services** — This is the area visible to client applications that connect to Virtual View Manager through JDBC/ODBC/SOAP.

- **My Home** — This area represents your (current user’s) own workspace.

- **Shared** — This area is shared by all users in the system.

  - In My Home and Shared, you can create resources, which are described in the chapter Introduction.

  - **examples** — This is a folder containing some sample resources to help you start using Virtual View Manager.

- **<host machine>** — This area reflects the machine that hosts the server.

As you create resources for your modeling, the tree structure will expand.

For further details on the resource tree, see the Virtual View Manager chapter in the User Guide.
Example Resources

Some resources have been pre-created and are displayed in the resource tree. Only those users with administrative privileges can delete these resources. Others can view them and execute them, as described in the following sections.

Steps to view the example resources
1. Expand the Shared > examples folder in the resource tree, and expand each node to see what is available.
2. Expand the Virtual View Manager Data Services > Databases > examples folder in the resource tree, and expand each node to see what is available.

Steps to open a resource
1. Right-click the desired resource, and select Open.
2. In the case of a leaf-node, you can also double-click it.

The example resources are described in the following sections. Getting Started with Virtual View Manager is a tutorial that describes how to create some of these resources and use them to solve business problems.

Sample Data Sources

Three underlying, physical data sources (orders, inventory, and productCatalog.xml) containing information about a company’s business with its suppliers and customers have been examined, and resources from them have been selected and added to the IBM® Cognos® Virtual View Manager metadata repository.

These data sources are displayed in the resource tree with their corresponding user-defined names ds_inventory, ds_orders, and ds_XML within Desktop > Shared > examples. When you expand a data source node, you see its contents.

Relational table data sources are indicated by this icon: 
File data source are indicated by this icon: 

When you double-click a table in a data source or an XML file, an editor opens on the right displaying the corresponding schema.

To view the contents of a table, right-click the table name in the resource tree, and select Show Contents.

ds_inventory
This data source has four tables with information on a company’s inventory and transaction with their suppliers, as described here:
• inventorytransactions — contains transaction details between the company and its supplier
• products — contains information about all the products in the company
• **purchaseorders** — contains details about the products the company has purchased from the supplier

• **suppliers** — contains information on suppliers and their contact information

### ds_orders
This data source has four tables with information on customers and their orders, as described here:

• **customers** — contains information on customers and their contact information

• **orderdetails** — contains information about every order from customers

• **orders** — contains information on all the orders from customers

• **shippingmethods** — contains information on shipping ID and shipping methods

### ds_XML

This data source has one file:

• **productCatalog.xml** — contains product category information in XML format. In order to query an XML data source, we need to transform the XML data into tabular form. Chapter 1, Introduction contains information on transformation (see Transformations).

For further details on data sources, see the Data Sources chapter in the User Guide.

### Sample Transformations

Two sample transformations are displayed in the resource tree:

• **getInventoryTransactions** — transforms tabular data from different sources into XML form using the definition set InventoryTransactions.

• **productCatalog_Transformation** — transforms the XML data in ds_XML into tabular form.

**Note:** The tutorial in this guide does not use getInventoryTransactions.

The sample transformations are displayed in the resource tree under Desktop > Shared > examples. Double-click the productCatalog_Transformation in the resource tree. Its schema opens on the right in the schema editor.

Click the **Execute** button on the toolbar to see the output of the executed transformation.

See the chapter Procedures in the User Guide for details on transformations.

### Sample Views

Four sample views—FederatedView, ViewOrder, ViewSales, and ViewSupplier—are displayed in the resource tree. These views are unpublished in the sense that they are not ready for external client access.

The sample views are described next.

See the chapter Views in the User Guide for details on views.
**FederatedView**
This view provides a unified, federated view of other views, ViewOrder, ViewSales, and ViewSupplier. It examines the three separate views and joins them on one column ProductID.

Double-click **FederatedView** in the resource tree to see the tables and their relationships.

Click the **Execute** button (1) to see the result in the lower pane.

Click the **Next** button (2) to scan all the rows of data retrieved.

**ViewOrder**
This view provides a specific view of the ds_orders data source and retrieves details about orders and customers. It examines three tables—customers, orders, and orderdetails—from the ds_orders data source, and joins the tables on two columns, CustomerID and OrderID. Then, it selects specific columns for projection in the execution result set.

Double-click **ViewOrder** in the resource tree and click the **Execute** button (1) on the toolbar to see the result.

**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 productCatalog_transformation, and joins them on one column, ProductID. Then, it selects specific columns for projection in the execution result set.

Double-click **ViewSales** in the resource tree and click the **Execute** button (1) on the toolbar to see the result.

**ViewSupplier**
This view provides a specific view of the ds_inventory data source and retrieves information about inventory and suppliers. It examines three tables—inventorytransactions, purchaseorders, and suppliers—from the ds_inventory data source, and joins the tables on two columns, PurchaseOrderID and SupplierID. Then, it selects specific columns for projection in the execution result set.

**Sample Definition Set ( )**
An XML-type definition set, InventoryTransactions, has been pre-created. The tutorial in this guide does not use this definition set. However, you can use it at any time to create other resources such as a transformation similar to getInventoryTransactions.

See the chapter **Definition Sets** in the **User Guide** for details on definition sets.

**Sample SQL Script**
An SQL script, LookupProduct, has been pre-created. The tutorial in this guide does not use this script. However, you can use it at any time in other resources, for example the sample transformation named getInventoryTransactions.

See the chapter **Procedures** in the **User Guide** for details on SQL Script.
Sample Published View

A resources in Desktop > Virtual View Manager Data Services are considered to be published.

One sample view, FederatedView, has also been made available to client applications that use JDBC/ODBC to connect to the server. You can find this published view within Desktop > Virtual View Manager Data Services > Databases > examples.

Right-click the published FederatedView in the resource tree (in Desktop > Virtual View Manager Data Services > Databases > examples) and select Show Contents. The columns in this view are shown, and the result of running the query is shown on the right.

Select the Info tab in the editor on the right to view the information about this published resource.
Chapter 3: Sample Resources
Chapter 4: Getting Started with IBM Cognos Virtual View Manager

This chapter provides a step-by-step tutorial that describes how to use the Modeler in IBM® Cognos® Virtual View Manager to create data resources.

About this Tutorial

This tutorial teaches you how to build a unified solution to address a typical business scenario. You learn how to perform the following data modeling tasks:

- Examine the underlying, physical data sources and select only the resources you want for your model.
- Join tables, and provide aliases for column names.
- Build and execute views.
- Publish views as Virtual View Manager database tables.

See "Tutorial Work Flow" (p. 23) to understand why you need to perform this sequence of tasks.

For more information about the Modeler, see the IBM® Cognos® Virtual View Manager User Guide.

Tutorial Scenario

For the example scenario in our tutorial, we use ALPHA Manufacturing, a fictitious company that distributes a wide variety of high technology widgets. These widgets have a variety of complex manufacturing and order fulfillment times along with a range of associated margins to ALPHA. ALPHA uses three data sources to store data about customers, products, orders, suppliers, and purchases. ALPHA’s three departments—Order, Purchase, and Sales—use these three data sources for their transactions.

Using IBM® Cognos® Virtual View Manager, you can create a unified view across all of these data sources. When people from various departments at ALPHA work independently on solving a common problem, such as the one described next, they need only access Virtual View Manager to gather all the information they need.

The scenario addressed by our tutorial is as follows:

- A customer, Landmark Systems, contacts ALPHA’s Order department and places an order for ALPHA’s product Widget 5 on 02/06/03.
- ALPHA’s Order department promises to deliver the product by 02/13/03.
- The customer, Landmark Systems, doesn’t receive the product on 02/13/03 and therefore contacts ALPHA’s Service department on 02/14/03.
• ALPHA’s Service department gets the order ID, order date, and product name from the customer.
• ALPHA starts the investigation with the information obtained from the customer.
• ALPHA’s Order department finds the product status to be open.
• ALPHA’s Purchase department tracks down the problem to the supplier:
  • ALPHA required the product from the supplier on 2/10/03.
  • Supplier promised to deliver the product on 2/10/03.
  • Supplier did not ship the product to ALPHA until 2/12/03.
  • Additionally, supplier did not use a fast shipping method.
• ALPHA’s Sales department finds out that the lead time on the order is one day which was not honored by the supplier.
• ALPHA’s Service department contacts the supplier, reports the problem, and negotiates a price reduction.
• ALPHA’s Service department contacts the customer, apologizes for the delay, and gives a higher discount on the order.

Data Sources Where Information Is Stored
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. See Tables and File in this Example next for the list of tables you’ll select from these data sources. The third data source is a product catalog. It has information about ALPHA’s products including prices, reorder level, and lead time, all of which is stored in an XML data source named product-Catalog.xml.

Tables and File in this Example
For this example, you will use the following tables from the inventory data source:
• inventorytransactions
• products
• purchaseorders
• suppliers
You will use the following tables from the orders data source:
• customers
• orderdetails
• orders
• shippingmethods
Tutorial Work Flow

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. For your convenience, this example shows how to build the unified view progressively in the following sequence of tasks:

- Add data sources to IBM® Cognos® Virtual View Manager (see "Adding Data Sources" (p. 23)). You will add the three underlying data sources to the Virtual View Manager metadata repository so you can query them. Each of these data sources will display as a data source in the resource tree.

- Create three simple views to retrieve data from the underlying data sources (see "Building Simple Views" (p. 30)). Each of these views will extract specific information for you to address the business issue at hand.

- Create the final unified view by combining the three individual views (see "Creating a View" (p. 37)). This view will unify the separate pieces of information you retrieve through the three individual views you create.

- Create a Virtual View Manager database. This database will be visible to client applications using JDBC/ODBC to connect to the Virtual View Manager server. See "Publishing Views to IBM Cognos Virtual View Manager Databases" (p. 40).

- Publish the unified view (see "Creating a View" (p. 37)). This unified view will provide a single view solution for the business issue at hand.

Adding Data Sources

Adding a data source means creating a IBM® Cognos® Virtual View Manager representation of the actual, underlying data source in the Virtual View Manager 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 Virtual View Manager 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 the Virtual View Manager server.

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

For details on privileges, see the chapter Security in the User Guide.

This section describes how to add the sample data sources—orders, inventory, and productCatalog.xml—to a folder named sources in Desktop/Shared.

Steps to create a folder named sources within Desktop/Shared

1. Right-click Shared, and select New Folder.
2. Type sources for the name in the input field, and click OK.
The newly created folder is added to the resource tree under Shared.

**Steps to add the orders data source**

1. Right-click the sources folder and select New Data Source.
   
   The New Physical Data Source window appears displaying a list of drivers. Here, you need to select the driver that will connect your data source to the Virtual View Manager server. Use the IBM® Informix® driver to connect to the orders data source you will use for this example.

2. Scroll down if necessary, select Informix 9.x for the driver, and click Next.
   
   The window for specifying the data source information appears.

3. Select the Basic tab if it is not already selected.
   
   - Type a unique name for the data source in the Name field, for example ds_orders. 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 Virtual View Manager representation of the underlying data source orders.
   
   - In the Host field, type the name or IP address of the computer where the Virtual View Manager server is installed. If the server is installed on your local machine, you can type the computer name or localhost.
   
   - In the Port field, type the port number. The default port number is 9408 for the IBM Informix database that is provided with Virtual View Manager. If you use an external database server, ensure the port number is correct.
   
   - In the Server field, type VVMRepository.
   
   - Type orders in the Database Name field. Virtual View Manager uses this entry to locate the underlying data source orders.
   
   - Type tutorial in the Login and Password fields. These entries are the user name and password to access the underlying data source. For example, if using IBM Informix, these values are the username and password you provided during the installation.
   
   - For now, skip the following fields: Save Password, Pass-through Login, and Transaction Isolation.
     
     For details on these properties, see the Data Sources chapter in the User Guide.

4. Select the Advanced tab.
   
   - Click JDBC Connection Properties.
   
   - Click the plus sign beside Add JDBC Connection Properties.
   
   - Type DB_LOCALE in the Property field.
- Type EN_US.utf8 in the Value field.
- Leave the remaining fields set to their default values.
- Click OK.

5. Click Next.
   The window for selecting the resources from the underlying data source opens.

6. In the tree on the left, expand ds_orders, orders, and informix.

7. Select the check boxes for customers, orderdetails, orders, and shipping methods.
   These tables contain the information you need for customer contact and order details.

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

**Steps to view the data source schema: (optional)**
1. Expand Shared > sources > ds_orders > orders > informix.
   Each node represents a table you selected in Step 6, "Steps to add the orders data source" (p. 24).
   All of these tables are available for your modeling in Virtual View Manager.

2. Double-click the customers table name.
   The table editor opens on the right, displaying the columns and their respective data types.

**Steps to add the inventory data source**
1. Right-click the sources folder and select New Data Source.
   The New Physical Data Source window appears displaying a list of drivers. Here, you need to select the driver that will connect your data source to the Virtual View Manager server. Use the IBM® Informix® driver to connect to the orders data source you will use for this example.

2. Scroll down if necessary, select Informix 9.x for the driver, and click Next.
   The window for specifying the data source information appears.

3. Select the Basic tab if it is not already selected.
   - Type a unique name for the data source in the Name field, for example ds_inventory.
     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 Virtual View Manager representation of the underlying data source orders.
   - In the Host field, type the name or IP address of the computer where the Virtual View Manager server is installed.
     If the server is installed on your local machine, you can type the computer name or localhost.
   - In the Port field, type the port number.
The default port number is 9408 for the IBM Informix database that is provided with Virtual View Manager. If you use an external database server, ensure the port number is correct.

- In the Server field, type VVMRepository.
- Type inventory in the Database Name field.
  Virtual View Manager uses this entry to locate the underlying data source orders.
- Type tutorial in the Login and Password fields.
  These entries are the user name and password to access the underlying data source. For example, if using IBM Informix, these values are the username and password you provided during the installation.
- For now, skip the following fields: Save Password, Pass-through Login, and Transaction Isolation.
  For details on these properties, see the Data Sources chapter in the User Guide.

4. Select the Advanced tab.
   - Click JDBC Connection Properties.
   - Click the plus sign beside Add JDBC Connection Properties.
   - Type DB_LOCALE in the Property field.
   - Type EN_US.utf8 in the Value field.
   - Leave the remaining fields set to their default values.
   - Click OK.

5. Click Next.
   The window for selecting the resources from the underlying data source opens.

6. In the tree on the left, expand ds_inventory, inventory, and informix.

7. Select the check boxes for inventorytransactions, products, purchaseorders, and suppliers.
   These tables contain the information about product inventory transactions, purchase order details, and supplier contact information.

8. Click Finish.
   You can expand sources in the resource tree to see whether the data source ds_inventory is added. ds_inventory is now ready for your use. If you don’t see the newly added data source in the resource tree, right-click the Desktop node, select Refresh, and then look for it in sources.

Expand Shared > sources > ds_inventory. Each node within ds_inventory represents a table you selected in Step 6. Expand each node to see the columns in that table.

All of these tables and columns are available for your modeling in Virtual View Manager.
If you want to view the schema of this data source, follow the steps described for viewing the orders data source schema in Steps to view the data source schema: (optional).

**Steps to add the XML data source**

1. Navigate to Shared > sources in the resource tree, right-click sources, and select New Data Source.
   
   The name of the XML data source used here is productCatalog.xml, which is located in the directory where Virtual View Manager is installed. By default, it is located in the `installation_location/docs/examples` directory.

2. In the Add Physical Data Source window, select File-XML for the driver to connect to the XML file, and click Next.
   
   Virtual View Manager displays a dialog box for you to provide the connection information for this type of data source.

3. In the screen for providing connection details:
   
   - In the Name field, type a unique name for the XML file data source, for example, ds.XML. This name is user-defined and will display in the resource tree when the process of adding the data source is complete.
   
   - Select Local File System and use the Browse button to locate the path to the XML file.
     
     Note that the root path does not include the name of the XML file, and only points to the directory where the file is. The file is located in the `installation_location/docs/examples` directory.
     
     You can also type the root path to the XML file in the Root Path field, instead of using the Browse button.
   
   - For now, skip the Character Set, Schema Location, and No Namespace Schema Location fields.
     
     You can get the details of this field in the Data Sources chapter in the User Guide.
   
   - Make sure that the value in the File Name Filter(s) field is: *.xml, and click Next.
   
   - Expand the ds.XML node, and select productCatalog.xml, which is the desired data source, and click Finish.
     
     The ds.XML data source is added to the resource tree at Shared > sources.

**Summary**

So far, you’ve performed the following tasks:

- Added two IBM® Informix® data sources—orders and inventory—to the Virtual View Manager server.
- Selected the following tables from the orders data source:
  
  customers
orderdetails
orders
shippingmethods

- Selected the following tables from the inventory data source:
  inventorytransactions
  products
  purchaseorders
  suppliers

- Added one XML data source which has one file named productCatalog.xml.

What’s Next After Adding the Data Sources?

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

The process of converting hierarchical data into tabular form is called transformation in the system. See also Tabular and Hierarchical Data and Transformations.

The following section describes how to convert XML data into tabular form.

Transforming XML Data in the XML Data Source

This section gives quick steps for transforming our XML data source. For further details on transformation, see the Procedures chapter in the User Guide.

Steps

1. Right click Shared > sources, and select New Transformation.
2. In the Create Transformation window, select XSLT Transformation, and click Next.
3. In the New Transformation dialog, type a unique name in the Transformation Name field for the transformation such as productCatalog_transformation.
4. Expand Shared> sources > ds_XML, and select productCatalog.xml.
5. Click Finish.

The transformation is added to the resource tree under Shared > sources.

Selecting Output Columns in the XML Transformation

When the transformation of the XML data source is added to the resource tree, the editor opens on the right. The XML schema from the productCatalog.xml data source is displayed in its transformation (productCatalog_transformation) under the name catalog.
At this point, you select the columns that you need for your analysis as described in the following steps.

**Steps**

1. Expand all the nodes in catalog so that you have the following picture:

   ![](image1)

   Tip: To widen the pane where the catalog nodes are displayed, use the Source and Target column headers.

2. Select CategoryName on the left, and click the Create Link and Target button on the toolbar.

   An entry is created with the name CategoryName under outputs in the Target column on the right, and the source and target are linked by a line.

   ![](image2)

3. Select other columns, and click Create Link and Target.

   After selecting all the columns, the screen should look as follows:

   ![](image3)
Tip: For consecutive multi-selection, press the Shift key and make your selection. Otherwise, press the Ctrl key and make your selection.

4. Select ProductID, ProductName, and CategoryID one by one on the right, and select the upward triangle (↑) on the toolbar to move the entries up, as shown next:

5. Click the disk icon (حفظ) on the toolbar to save the transformation.

6. Click the × on the tab to close the productCatalog_Transformation.

Data Sources and Transformation Are Ready for Use

The data sources ds_inventory and ds_orders, and the transformation productCatalog_transformation are now ready for your use.

You will query these sources to investigate the issue at hand for ALPHA, namely 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 three departments—Order, Purchase, and Sales—at ALPHA.

Building Simple Views

This section describes how to create folders for storing your views in an organized fashion, 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 source(s) and retrieves the desired data.

For additional information on building and executing views, refer to the Views chapter in the User Guide.

In this section, you create three views as the three departments at ALPHA would do as follows:

- View 1 to obtain information on the order and customer (Order department).
- View 2 to obtain information on the purchase order and supplier (Purchase department).
- View 3 to obtain product sales information (Sales department).
**View 1: Obtaining Order Information**

For this view, 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.

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

**Steps**

1. Right-click **Shared > sources**, and select **New View**.

2. In the **New View** window, type **ViewOrder** as the name for the view and click **OK**.

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

   ![ViewOrder](image)

   Note that the editor has several tabs:
   - **Model** is where you assemble the tables.
   - **Grid** is where you specify query constraints and the columns to include in the output when you execute your view.
   - **SQL** is where the SQL for the view is displayed when you design the view in the **Model** and **Grid** panels. You can also type SQL in the **SQL** tab, but here we’ll use the **Model** and **Grid** tabs to build the view.
   - **Columns** lists the columns you select for projection in the view execution result.
   - **Indexes** does not actually index anything; it reports the index in the system tables, if the view is published.
   - **Foreign Keys** provides a place for you to define foreign keys.
   - **Caching** provides an entry point to access the cache setting panel where you can configure the cache for storing the execution result of the current resource.
   - **Info** is where you can provide annotations on the view.

3. Expand **Shared > sources** in the resource tree, and from the ds_orders data source drag the following tables into the **Model** panel in the editor:
   - orders
4. Click OrderID in orders and drag it onto OrderID in orderdetails. A line appears representing the inner join between the two tables.

5. Join CustomerID in customers with CustomerID in orders. The tables are joined as shown below.

For details on joins, see the section *Designing a View* in the chapter *Views* in the User Guide.

6. In 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. In this step, we will limit the columns we want in the result set.

   - Click the first cell under Column and select orderdetails.OrderID in the drop-down list.
   - Click more rows, as needed, 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

7. In the Alias column, click the cell corresponding to customers.ContactFirstName and type an alias, for example CustomerContactFirstName. Press the Enter key after typing the alias.
An alias makes a column name unique, which is necessary to avoid conflicts with columns from other data sources having the same name.

8. Type `CustomerContactLastName` as the alias for customers.ContactLastName and `CustomerContactPhone` as the alias for customers.PhoneNumber, respectively.

The next screen illustrates how the table columns are specified in the `Grid` panel.

Note the asterisks next to the View Order, Model, and Grid tab names, indicating that you’ve made changes but not saved them.

9. Save the view by clicking the disk icon ( ) on the toolbar.

10. Optionally, you can view the SQL for this view by selecting the `SQL` tab. Once you’ve viewed the SQL, return to the modeling area by selecting the `Model` tab. Typing or editing the SQL in the `SQL` panel will invalidate the design made in the `Model` and `Grid` panels, which you don’t want to do in this tutorial.

11. Execute the view by clicking the `Execute` button ( ) in the editor toolbar.

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

12. 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

   Note that 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 next screen illustrates how the view execution results are displayed in the `Result` panel.
13. Click the on the tab to close the ViewOrder view.

Next, you create and build View 2 to pursue what happened in the Purchase department with respect to the shipment from the supplier.

**View 2: Getting Supplier Information**

For this view, 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.

Therefore, you include the tables inventorytransactions, purchaseorders, and suppliers in this view.

**Steps**

1. Right-click Shared > sources in the resource tree, and add a view named ViewSupplier.
   
   See "View 1: Obtaining Order Information" (p. 31) for information about the tabs in the view editor.

2. Drag the following tables from ds_inventory and drop them into the Model panel of ViewSupplier:
   
   inventorytransactions
   purchaseorders
   suppliers


4. Join SupplierID in suppliers with SupplierID in purchaseorders.

5. In the Grid panel, select the following columns as described in Step 6 in the "View 1: Obtaining Order Information" (p. 31) for information about the tabs in the view editor.

   inventorytransactions.ProductID
   inventorytransactions.TransactionID
   purchaseorders.DateRequired
   purchaseorders.DatePromised
purchaseorders.ShipDate
purchaseorders.SupplierID
suppliers.SupplierName
suppliers.ContactName
suppliers.PhoneNumber

6. Provide aliases for suppliers.ContactName and suppliers.PhoneNumber, by typing SupplierContactName and SupplierPhoneNumber in the Alias column, respectively.

Recall that an alias makes a column name unique, which is necessary to avoid conflicts with SupplierContactName and SupplierPhoneNumber columns from other data sources having the same name.

7. Save the view (and its SQL).

8. Execute the view to see the result.

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

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

10. Click the on the tab to close the ViewSupplier view.

Now, you will proceed to build View 3 to pursue what the Sales department can find out about the product and the lead time for delivery.
**View 3: Obtaining Sales Information**

For this view, you use the orders data source and the transformation productCatalog_transformation.

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

**Steps**

1. Right-click Shared > sources in the resource tree, and add a view named ViewSales.
   See "View 1: Obtaining Order Information" (p. 31) for information about the tabs in the view editor.

2. Expand the ds_orders node under Shared > sources and drag the orderdetails table into the Model panel.

3. From under Shared>sources, drag the productCatalog_transformation into the Model panel.

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

5. In the Grid panel, select the following columns:
   - productCatalog_transformation.
   - orderdetails.Status
   - orderdetails.Discount

6. Save the view (and the SQL).

7. Execute the view.

8. In the result set displayed in the Result panel, locate one of the three rows in which ProductID = 23 which look as follows:
   - ProductID: 23
   - ProductName: Widget 5
CategoryID: 7
CategoryName: Networking
ProductDescription: WidgetModel
SerialNumber: 5
ReorderLevel: 5
LeadTime: 1 Day
Status: open
Discount: 0.05

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

9. Click the X on the tab to close the ViewSales view.

Summary

After adding the data sources to the Virtual View Manager 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.

What’s Next After Building Three Separate Views?

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

Creating a View

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

Steps

1. Expand Shared > sources in the resource tree, and add a view named FederatedView.

2. Drag the following views and drop them into the Model panel of FederatedView:
5. In the **Grid** panel, select the following columns:
   - ViewOrder.*
   - ViewSales.ProductName
   - ViewSales.LeadTime
   - ViewSupplier.TransactionID
   - ViewSupplier.DateRequired
   - ViewSupplier.DatePromised
   - ViewSupplier.ShipDate
   - ViewSupplier.SupplierID
   - ViewSupplier.SupplierName
   - ViewSupplier.SupplierContactName
   - ViewSupplier.SupplierPhoneNumber
6. Save the view (and its SQL).
7. Execute the view. Result rows 1-50 are displayed.
8. 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.
   **Tip:** Click Next (>) in the **Result** panel to browse all the rows.
   The desired 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

Note that 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.

FederatedView is the single, unified view that represents your business solution.

9. Click the on the tab to close FederatedView.

**What’s Next After Building the Federated View?**

Optionally, you can publish each view you have created so far. Publishing a view means, making it available to client applications that connect to the server via JDBC/ODBC. To do so, you must publish a view as a Virtual View Manager database table. Once you publish a view, you can re-use it as a relational database table to query further.

*For further details on publishing, see the Publishing Resources chapter in the User Guide.*

**Publishing Your Views**

The location to publish a view for JDBC/ODBC client applications is IBM® Cognos® Virtual View Manager Data Services on your Desktop.

Virtual View Manager data services are the entry points for external applications to communicate with the Virtual View Manager 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 Virtual View Manager data service of the type Virtual View Manager 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 Virtual View Manager data service of the type Virtual View Manager database.

**Steps**

1. Right-click Virtual View Manager Data Services > Databases in the resource tree, and select New Virtual View Manager Data Service.

    The Add Virtual View Manager Data Service window appears prompting you to enter a name for the service.
2. In the Data Service Name field, type cds_tutorial as the unique name for the Virtual View Manager data service you are creating, and click OK.

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

The next section describes how to publish your views to the newly created Virtual View Manager database (cds_tutorial).

### Publishing Views to IBM Cognos Virtual View Manager Databases

This section describes how to publish the federated view you’ve created using the steps provided in "Creating a View" (p. 37). The process is similar for publishing other resources.

**Steps**

1. Expand Shared > sources in the resource tree.

2. Right-click FederatedView, and select the Publish option.

3. In the Publish window, specify the location to publish the view:
   - Navigate to Virtual View Manager 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. However, we won’t create catalogs or schemas here.
   - You can accept the default name displayed in the Name field, and click OK.

   The view is now published.

You can expand Virtual View Manager Data Services > Databases > cds_tutorial in the resource tree to see the view published as FederatedView.

If you would like, you can publish the other views in the same manner as you did in this section.

### Summary

This chapter has provided an exercise for you to start using the Modeler in IBM® Cognos® Virtual View Manager to build a sample business solution.

You performed the following tasks:

- Added data sources to the Virtual View Manager metadata environment.
- Created three separate views to retrieve specific information about ALPHA’s customer, order, order date, promised date of product delivery to the customer, supplier, and date of product delivery from the supplier.
- Created a single federated view combining the three individual views.
- Published the federated view.

### Salient Features

- During this exercise you created three separate views.
Each view represents the activities of a particular department (Order, Purchase, or Sales) at ALPHAManufacturing. In real world, that’s how things are likely to happen in an organization — each department conducting its own business without having to share information with another department. As you can see, the views represent internal organizational activities.

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 built the final, unified view from the three individual views you had created.

At ALPHAM, the Service department can make use of this view to render satisfactory service to the customer who originally reported a problem.

You learned how to use a view as a building block in its own capacity. Since you can use a Virtual View Manager database as any relational database, you can build any number or type of views you wish until a view matches your business need and publish that view as a Virtual View Manager database table. This task demonstrates the reusability of a view.

- You published the federated view.

You learned how to publish the federated view (FederatedView) you had created earlier 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.

**What’s Next?**

This guide has introduced the basic concepts surrounding IBM® Cognos® Virtual View Manager, and described the basic data modeling tasks such as building views and publishing views as Virtual View Manager database tables.

There are many more modeling and publishing tasks you can perform using Virtual View Manager, such as transforming data, creating procedures, and publishing views and procedures as Virtual View Manager databases and Web services.

Another important task is providing security to preserve data integrity. Each resource you created in this tutorial — data source, folder, view, and Virtual View Manager database — can be individually protected by granting security through specific access privileges to specific users.

For details on security and other modeling tasks, refer to the IBM® Cognos® Virtual View Manager User Guide and the on-line help topics in Virtual View Manager.

For details on the technologies underlying the creation of views and procedures, see the Reference Manual.
# Index

**A**
- audience of document, 5

**C**
- components, 7
- concepts, 7
- creating, 12
- custom Java procedures, 9

**D**
- data, 11
- databases, 10
- data services, 10
- data sources, 8, 12
  - sample, 16
  - tutorial, 22, 23, 28
- definition sets, 9
  - sample, 18
- description of product, 5

**E**
- example resources, 16

**F**
- folders, 11, 30

**H**
- hierarchical data, 11

**I**
- introspecting, 12

**L**
- logging in, 13

**M**
- modeling, 11, 15

**O**
- operations, 10

**P**
- packaged queries, 9
- parameterized queries, 9
- procedures, 8
  - creating, 12
  - custom Java procedures, 9
  - packaged queries, 9
  - parameterized queries, 9
- publishing, 12
- SQL scripts, 8
- transformations, 9
- published views
  - sample, 19
- publishing, 11, 12, 39, 40
- purpose of document, 5

**R**
- resources, 7, 10
  - example, 16
  - sample, 15

**S**
- sample data sources, 16
- sample definition sets, 18
- sample published views, 19
- sample resources, 15
- sample SQL scripts, 18
- sample transformations, 17
- sample views, 17
- security, 10, 11
- sets
  - definition, 18
- setting up, 13
- SQL scripts, 8
  - sample, 18

**T**
- tabular data, 11

Licensed Materials – Property of IBM