pentaho™

open source business intelligence™

Getting Started with Pentaho Data Integration

About This Document

If you have questions that are not covered in this guide, or if you find errors in the instructions or language, please contact the Pentaho Technical Publications team at documentation@pentaho.com. The Publications team cannot help you resolve technical issues with products.

Support-related questions should be submitted through the Pentaho Customer Support Portal at http://support.pentaho.com.

For information about how to purchase support or enable an additional named support contact, please contact your sales representative, or send an email to sales@pentaho.com.

For information about instructor-led training on the topics covered in this guide, visit http://www.pentaho.com/training.

Limits of Liability and Disclaimer of Warranty

The author(s) of this document have used their best efforts in preparing the content and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, express or implied, with regard to these programs or the documentation contained in this book.

The author(s) and Pentaho shall not be liable in the event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the programs, associated instructions, and/or claims.

Trademarks

Company Information

Pentaho Corporation
Citadel International, Suite 340
5950 Hazeltine National Drive
Orlando, FL 32822
Phone: +1 407 812-OPEN (6736)
Fax: +1 407 517-4575
http://www.pentaho.com

E-mail: communityconnection@pentaho.com

Sales Inquiries: sales@pentaho.com

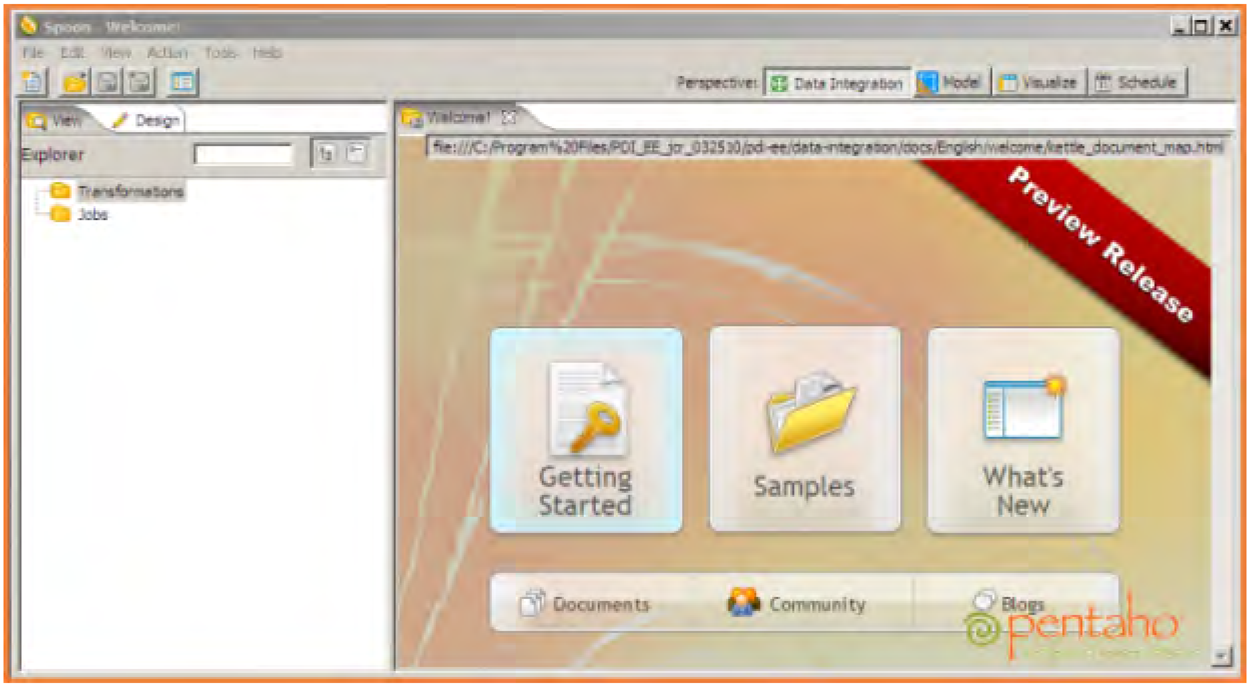
Documentation Suggestions: documentation@pentaho.com

Sign-up for our newsletter:
http://community.pentaho.com/newsletter/

Contents

Introduction

Thank you for choosing Pentaho Data Integration. Formerly known as Kettle, Pentaho Data Integration is a powerful Extraction, Transformation and Loading (ETL) solution based upon an innovative, metadata-driven approach. Pentaho Data Integration includes an easy to use, graphical design environment for building ETL jobs and transformations, resulting in faster development, lower maintenance costs, interactive debugging, and simplified deployment.



file:///C:/Program%20Files/PDI_EE_jcr_032510/pdi-ee/data-integration/docs/English/welcome/kettle_document_map.html

Common Uses

Pentaho Data Integration is an extremely flexible tool that addresses a broad number of use cases including:

- Data warehouse population with built-in support for slowly changing dimensions and surrogate key creation
- Data migration between different databases and applications
- Loading huge data sets into databases taking full advantage of cloud, clustered and massively parallel processing environments
- Data Cleansing with steps ranging from very simple to very complex transformations
- Data Integration including the ability to leverage real-time ETL as a data source for Pentaho Reporting

Key Benefits

Pentaho Data Integration features and benefits include:

- Installs in minutes; you can be productive in one afternoon
- 100% Java with cross platform support for Windows, Linux and Macintosh
- Easy to use, graphical designer with over 100 out-of-the-box mapping objects including inputs, transforms, and outputs
- Simple plug-in architecture for adding your own custom extensions
- Enterprise Data Integration server providing security integration, scheduling, and robust content management including full revision history for jobs and transformations

- Integrated designer (Spoon) combining ETL with metadata modeling and data visualization, providing the perfect environment for rapidly developing new Business Intelligence solutions
- Streaming engine architecture provides the ability to work with extremely large data volumes
- Enterprise-class performance and scalability with a broad range of deployment options including dedicated, clustered, and/or cloud-based ETL servers

Audience and Assumptions

This guide is written for IT managers, database administrators, and Business Intelligence solution architects who are new to Pentaho Data Integration.

You do not need any *specialized* experience, knowledge, or skills associated with ETL to follow the exercises in this evaluation guide; however, a basic understanding of ETL, SQL, and relational database concepts is helpful.

To complete the exercises, you must have an installation of Pentaho Data Integration 4.0 or later.

What This Guide Covers

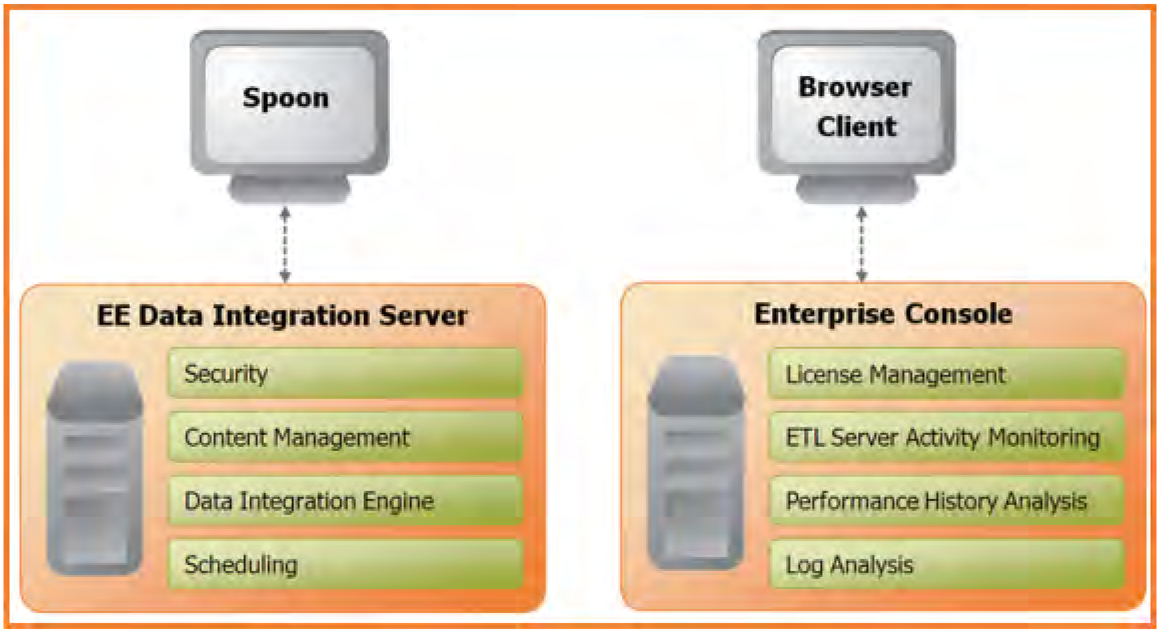
This guide walks you through the download and installation of Pentaho Data Integration and includes a few simple exercises that introduce you to the basic concepts required to build your first ETL transformation and job. The guide is organized into short modules allowing you to skip over any sections that do not interest you.

Module	Description	Estimated Time to Complete
Deployment Architecture	High level overview of the major components of Pentaho Data Integration and their functions	3 min.
Downloading and Installing Pentaho Data Integration	Where to download the latest version of Pentaho Data Integration and how to perform a basic installation	20 min.
Navigating the User Interface	Introduction to the Pentaho Data Integrations design client, Spoon	5 min.
Creating Your First Transformation	Learn how to add/configure transformation steps and connect them using hops. This example covers some of the most common ETL activities including reading source data, sorting, performing a lookup, and writing to a database.	30 min.
Running a Transformation	High level description of deployment and execution options including local, remote and clustered execution.	5 min.
Analyzing Results	Learn how to use the Execution Results pane for viewing logs, execution statistics and step metrics. This section also provides a brief introduction to the	10

Module	Description	Estimated Time to Complete
	interactive Preview and Debugging interface	
Building Your First Job	Learn how to build your first Job for coordinating common ETL related activities.	15
Scheduling	Introduction to scheduling execution of Transformations and Jobs using the Enterprise Edition Data Integration Server.	10
Building BI Solutions Using an Agile Approach	Introduction to using Spoon for quickly visualizing data sources and designing metadata to facilitate self-service BI solutions including reporting, ad hoc and OLAP analysis	20
Why consider Enterprise Edition	Learn about benefits getting a Pentaho Data Integration Enterprise Edition subscription.	Optional

Pentaho Data Integration Architecture

The diagram below depicts the the core components of Pentaho Data Integration Enterprise Edition



Spoon is the design interface for building ETL jobs and transformations. Spoon provides a drag and drop interface allowing you to graphically describe what you want to take place in your transformations which can then be executed locally within Spoon, on a dedicated Data Integration Server, or a cluster of servers.

Enterprise Edition (EE) Data Integration Server is a dedicated ETL server whose primary functions are:

Execution	Executes ETL jobs and transformations using the Pentaho Data Integration engine
Security	Allows you to manage users and roles (default security) or integrate security to your existing security provider such as LDAP or Active Directory
Content Management	Provides the ability to centrally store and manage your ETL jobs and transformations. This includes full revision history on content and features such as sharing and locking for collaborative development environments.
Scheduling	Provides the services allowing you to schedule and monitor scheduled activities on the Data Integration Server from within the Spoon design environment.

The **Enterprise Console** provides a thin client for managing deployments of Pentaho Data Integration Enterprise Edition including management of Enterprise Edition licenses, monitoring and controlling activity on a remote Pentaho Data Integration server and analyzing performance trends of registered jobs and transformations.

Downloading Pentaho Data Integration

Before you begin to download Pentaho Data Integration, you must:

- Have a data compression utility, such as 7Zip or WinZip to extract the downloaded files.
- Have *Java 6.0* already installed.

1. Go to the Pentaho Data Integration *download* page.
2. Fill out the contact form.
   You will receive a confirmation email that provides you with credentials to access the Pentaho Knowledge Base, which contains product documentation, support tips, and how-to articles.
3. Click the **Download Enterprise Edition** button.

Installing Pentaho Data Integration

The preview release of Pentaho Data Integration Enterprise Edition version 4.0 is a simple .zip (.tar for Linux and Macintosh) archive. To install, unzip **pdi-ee-4.0.0-preview.zip** using any standard compression utility. This creates a **pdi-ee** folder that contains the following files and directories:

File/Folder Name	Description
\data-integration	Contains the Spoon designer and command line utilities
\data-integration-server	Contains the data integration server including individual start/stop scripts
\enterprise-console	Contains the enterprise console server including individual start/stop scripts
license	Contains the scripts for installing a valid license.
getting_started_with_pdi.pdf	This document
start-servers.bat	Script file for starting all servers on Windows
start-servers.sh	Script file for starting all servers on Linux and Macintosh
stop-servers.bat	Script file for stopping all servers on Windows
stop-servers.sh	Script file for stopping all servers on Linux and Macintosh

Starting Pentaho Data Integration

The root directory of your Pentaho Data Integration installation, **\pdi-ee**, contains a set of scripts that make it convenient to start or stop all of the core server modules including the **Data Integration Server**, **Enterprise Console** server and **Hypersonic** (HSQLDB) database that contains the sample database used in the examples later in this guide.

Starting the Pentaho Data Integration Servers

To start Pentaho Data Integration servers...

1. Navigate to the folder where you have installed Pentaho Data Integration; for example, `...\Program Files\pdi-ee`.
2. Double-click **start-servers.bat** to start the servers.

Note: If you are using Linux or Macintosh, double-click **start-servers.sh**.

Starting the Spoon Designer

To start the Spoon designer:..

1. Navigate to the folder where you have installed Pentaho Data Integration; for example **c:\Program Files\pdi-ee**.
2. Double-click **launch-designer.bat** to start the servers.

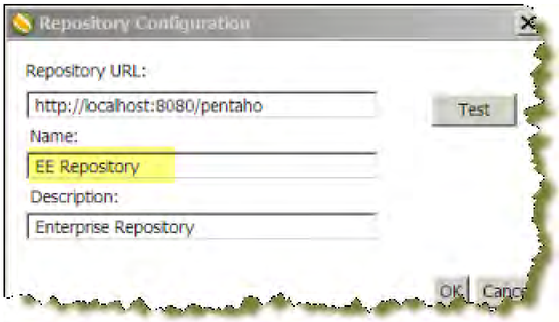
Note: If you are using Linux or Macintosh, double-click **launch-designer.sh**.

Connecting to the Enterprise Repository

Next, you will create a connection to the **Enterprise Repository** that is part of the **Data Integration Server**. The Enterprise Repository is used to store and schedule the example transformation and job you will create when performing the exercises in this document.

To create a connection to the Enterprise Repository:...

1. In the **Repository Connection** dialog box, click ⊕ (Add).
2. Select **Enterprise Repository:Enterprise Repository** and click **OK**.
   The **Repository Configuration** dialog box appears.
3. Enter a name and description (optional) for your repository connection. In the example below the name of the connection is **EE Repository**.



Repository URL: http://localhost:8080/pentaho

Name: EE Repository

Description: Enterprise Repository

4. Click **Test** to ensure your connection is properly configured. If you get an error, make sure you started your *Data Integration Server* .
5. Click **OK** to exit the **Success** dialog box.
6. Click **OK** to exit the Repository Configuration dialog box.
   Your new connection appears in the list of available repositories.
7. Log on to the Enterprise Repository by entering the following credentials: user name = **joe**, password = **password**.

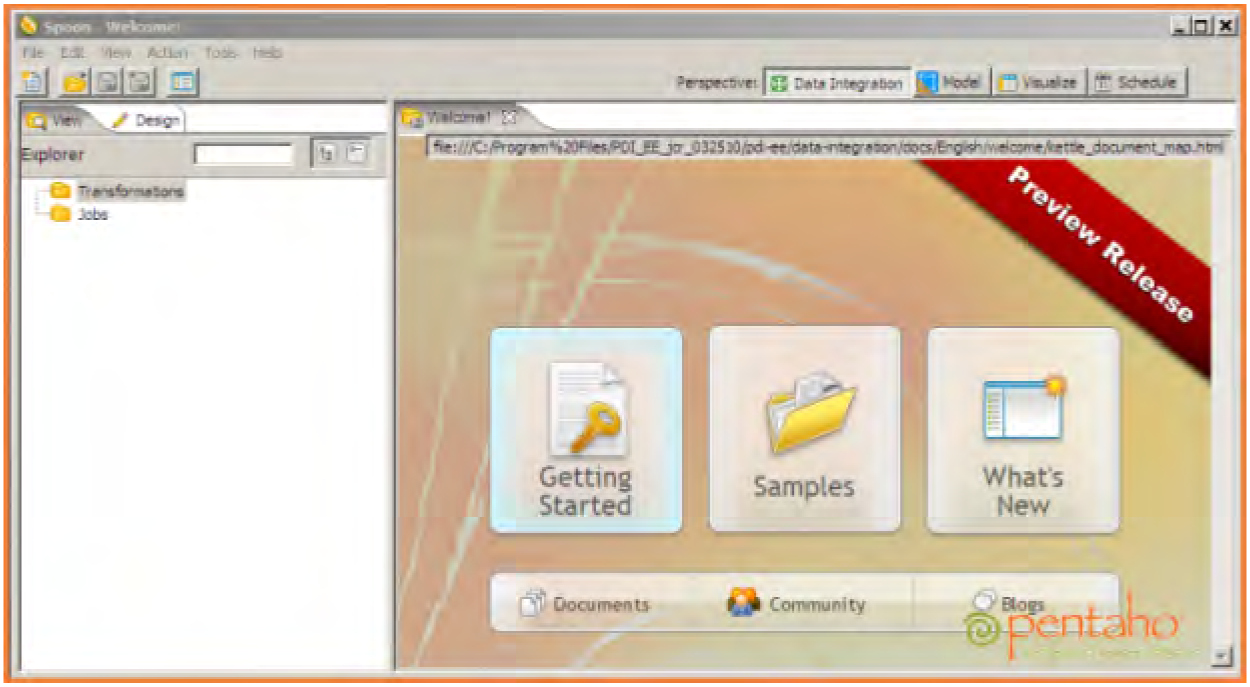
   The Data Integration Server is configured out of the box to use the Pentaho default security provider. This has been pre-populated with a set of sample users and roles including:

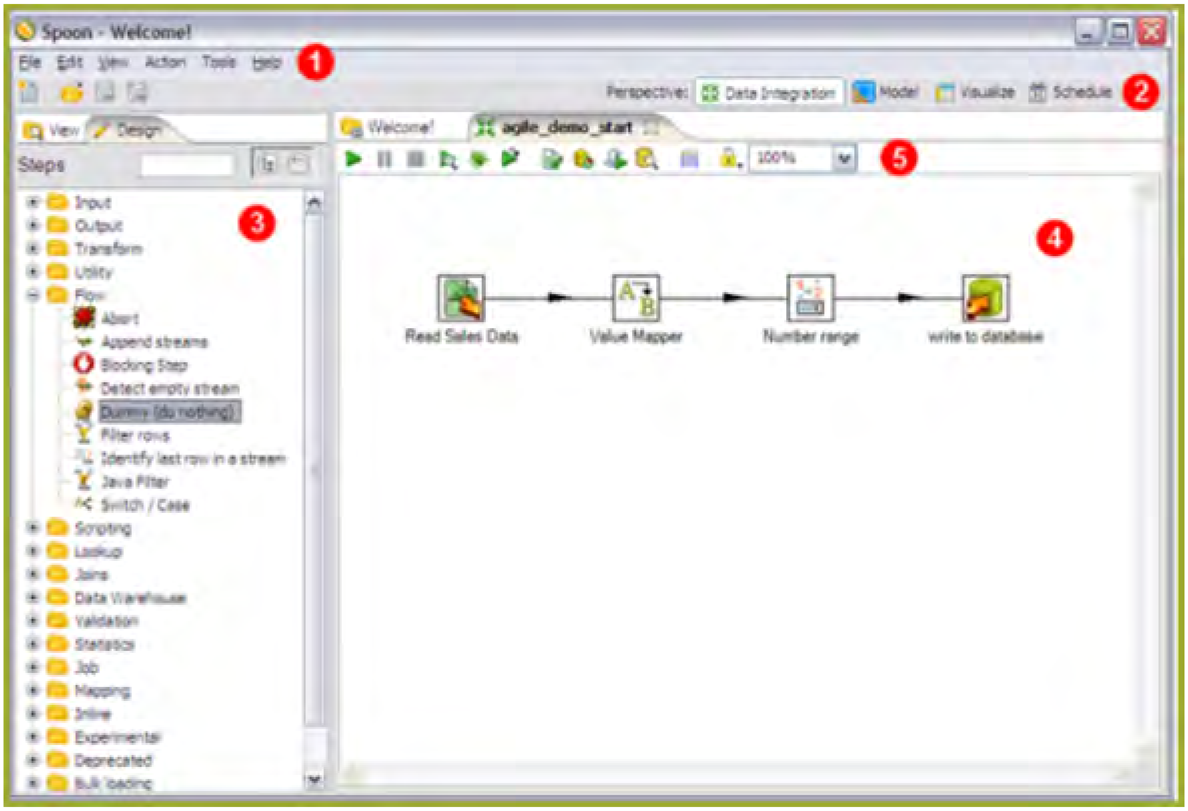
   - Joe — Member of the admin role with full access and control of content on the Data Integration Server
   - Suzy — Member of the CEO role with permission to read and create content, but not administer security

   Note: See the **Security Guide** available in the Pentaho Knowledge Base for details about configuring security to work with your existing security providers such as LDAP or MSAD.

Navigating through the Interface

The **Welcome** page contains useful links to documentation, community links for getting involved in the Pentaho Data Integration project, and links to blogs from some of the top contributors to the Pentaho Data Integration project.


Spoon - Welcome!
File Edit View Action Tools Help
Perspective: Data Integration Model Visualize Schedule
View Design
Explorer
Transformations
Jobs
Welcome!
file:///C:/Program%20Files/PDI_EE_tcr_032510/pdi-ee/data-integration/docs/English/welcome/kettle_document_map.html
Preview Release
Getting Started
Samples
What's New
Documents
Community
Blogs
pentaho



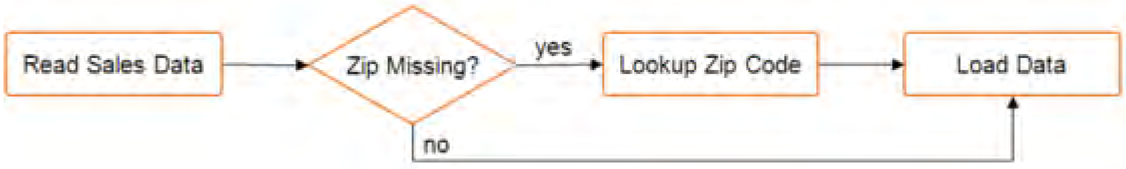
The Spoon Designer is organized into the components described in the table below:

Component Name	Description
1-Menubar	The Menubar provides access to common features such as properties, actions and tools
2-Main Toolbar	The Main Toolbar provides single-click access to common actions such as create a new file, opening existing documents, save and save as. The right side of the main toolbar is also where you can switch between perspectives: • **Data Integration** — This perspective (shown in the image above) is used to create ETL transformations and jobs • **Model** — This perspective is used for designing reporting and OLAP metadata models which can be tested right from within the Visualization perspective or published to the Pentaho BI Server • **Visualize** — This perspective allows you to test reporting and OLAP metadata models created in the Model perspective using the Report Design Wizard and Analyzer clients respectively • **Schedule** — This perspective is used to manage scheduled ETL activities on a Data Integration Server
3-Design Palette	While in the **Data Integration** perspective, the **Design Palette** provides an organized list of transformation steps or job entries used to build transformations and jobs. Transformations are created by simply dragging transformation steps from the Design Palette onto the Graphical Workspace, or canvas, (4) and connecting them with hops to describe the flow of data.
4-Graphical Workspace	The Graphical Workspace, or canvas, is the main design area for building transformations and jobs describing the ETL activities you want to perform.
5-Sub-toolbar	The Sub-toolbar provides buttons for quick access to common actions specific to the transformation or job such as Run, Preview and Debug.

Creating Your First Transformation

The Data Integration perspective of Spoon allows you to create two basic document types: transformations and jobs. Transformations are used to describe the data flows for ETL such as reading from a source, transforming data and loading it into a target location. Jobs are used to coordinate ETL activities such as defining the flow and dependencies for what order transformations should be run, or prepare for execution by checking conditions such as, "Is my source file available, or does a table exist in my database?"

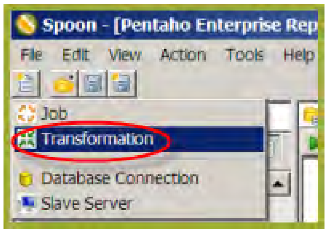
This exercise will step you through building your first transformation with Pentaho Data Integration introducing common concepts along the way. The exercise scenario includes a flat file (CSV) of sales data that you will load into a database so that mailing lists can be generated. Several of the customer records are missing postal codes (zip codes) that must be resolved before loading into the database. The logic looks like this:



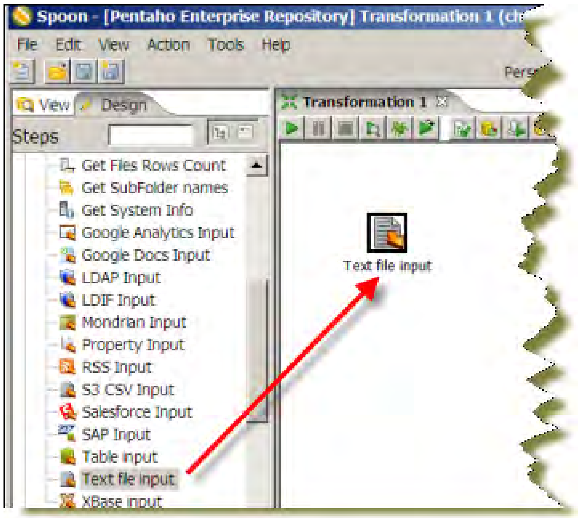
Retrieving Data from a Flat File

Follow the instructions below to retrieve data from a flat file.

1. Click (New) in the upper left corner of the Spoon graphical interface.
2. Select **Transformation** from the list.

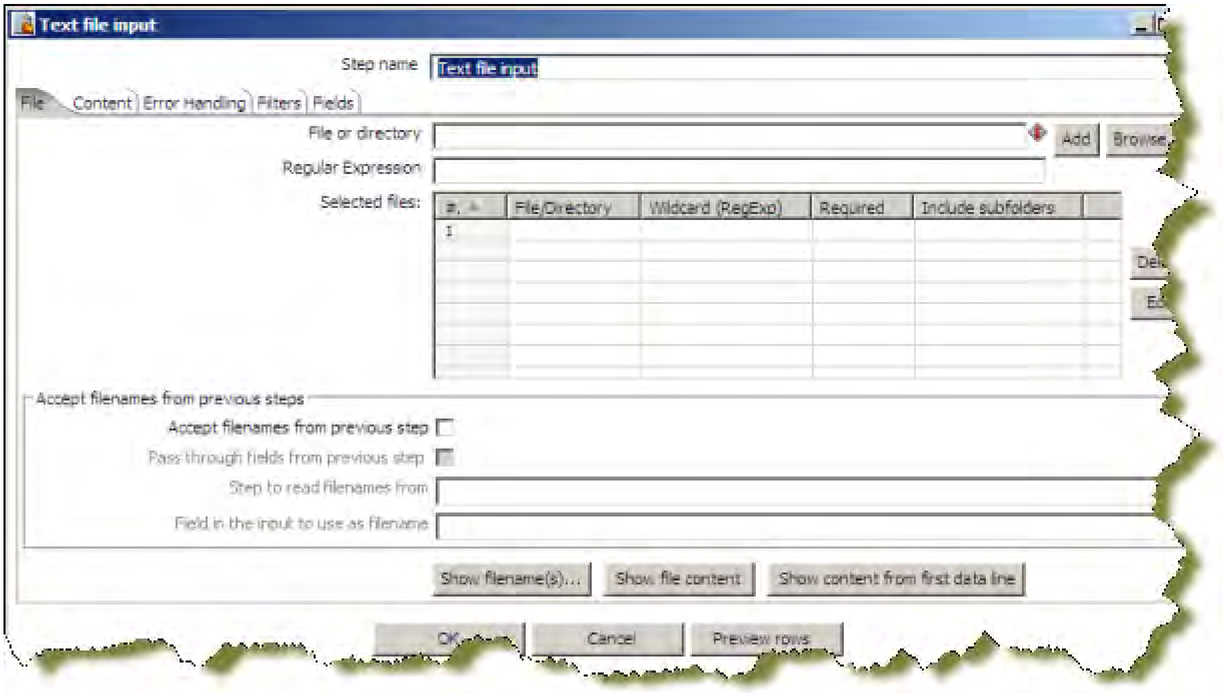

Spoon - [Pentaho Enterprise Rep
File Edit View Action Tools Help
Job
Transformation
Database Connection
Slave Server

3. Under the **Design** tab, expand the Input node; then, select and drag a **Text File** input step onto the canvas on the right.



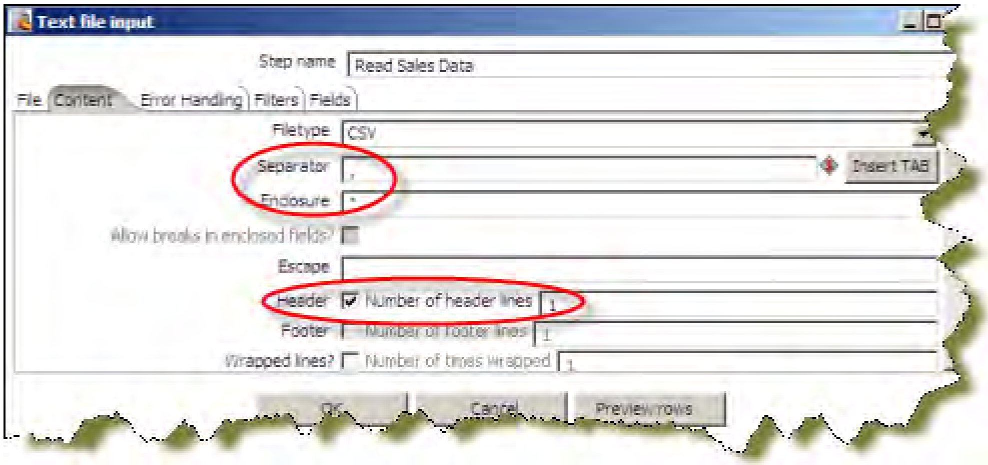
4. Double-click on the **Text File** input step.

The edit properties dialog box associated with the Text File input step appears. In this dialog box, you specify the properties related to a particular step.


Text file input

Step name: Text file input

5. In the **Step Name** field, type **Read Sales Data**.
   You are renaming the Text File Input step to Read Sales Data.
6. Click **Browse** to locate the source file, **sales_data.csv**, available at `...\pdi-ee\data-integration\samples\transformations\files`.
   The path to the source file appears in the **File or directory** field.
7. Click **Add**.
   The path to the file appears under **Selected Files**. You can look at the contents of the file by clicking the **Show file content** to determine things such as how the input file is delimited, what enclosure character is used, and whether or not a header row is present. In the example, the input file is comma (,) delimited, the enclosure character being a quotation mark (“) and it contains a single header row containing field names.
8. Click the **Content** tab.
   The fields under the **Content** tab allow you to define how your data is formatted.
9. Make sure that the **Separator** is set to comma (,) and that the **Enclosure** is set to quotation mark ("). Enable **Header** because there is one line of header rows in the file.


Text file input

Step name: Read Sales Data

Filetype: CSV

Separator: ,

Enclosure: "

Allow breaks in enclosed fields? □

Escape:

Header ☑ Number of header lines 1

Footer □ Number of footer lines 1

Wrapped lines? □ Number of times wrapped 1

10. Click the **Fields** tab and click **Get Fields** to retrieve the input fields from your source file.

A dialog box appears requesting that you to specify the number of lines you want to see, allowing you to determine default settings for the fields such as their format, length, and precision. Click **OK** and the summary of the scan results will be displayed. Click **Close** to return to the step properties editor.

11. Click **Preview Rows** to verify that your file is being read from correctly ;click **OK** to exit the step properties dialog box..
12. Click **File** -> **Save As** to save your transformation. Name your transformation, **Getting Started Transformation**.

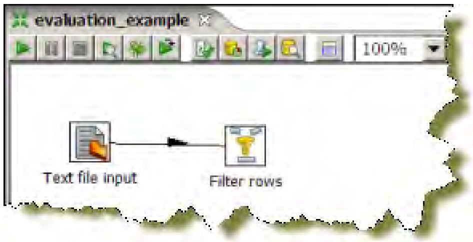
    Note: You can give your transformation any other name you want to use. Whenever you save a transformation, an **Enter a Comment** dialog box appears. The comment and your transformation are tracked for version control purposes in the Enterprise Repository.

Filter Records with Missing Postal Codes

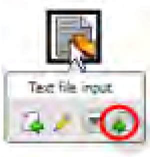
The source file contains several records with missing postal codes. You will now use the Filter Rows transformation step to separate out those records so that you can resolve them in a later exercise.

Your source file contains some rows that are missing zip codes. In this exercise, you are going to filter those records.

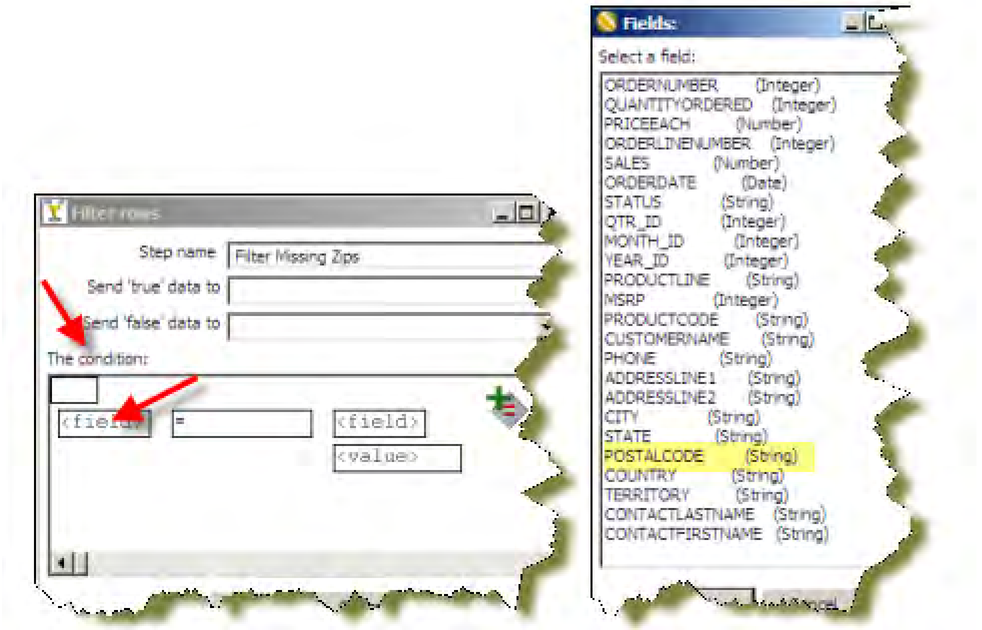
1. Add a **Filter Rows** step to your transformation. Under the **Design** tab, go to **Flow** -> **Filter Rows**.
2. Create a "hop" between the **Read Sales Data** (Text File Input) step and the **Filter Rows** step. Hops are used to describe the flow of data in your transformation. To create the hop, click the **Read Sales Data** (Text File input) step, then press the <**SHIFT**> key down and draw a line to the Filter Rows step.

    Alternatively, you can draw hops by hovering over a step until the hover menu appears. Drag the hop painter icon from the source step to your target step.

    Note: For more information on hops including a description of color coding and hop icons, see the *Pentaho Data Integration User Guide* in the Pentaho Knowledge Base; (document available after the GA release of Pentaho Data Integration).
3. Double-click the **Filter Rows** step.
   The **Filter Rows** edit properties dialog box appears.
4. In the **Step Name** field type, **Filter Missing Zips**.
5. Under **The condition**, click `<fields>`. A dialog box that contains the fields you can use to create your condition appears.


Filter rows

Step name: Filter Missing Zips

Send 'true' data to:

Send 'false' data to:

The condition:

<field> = <field>

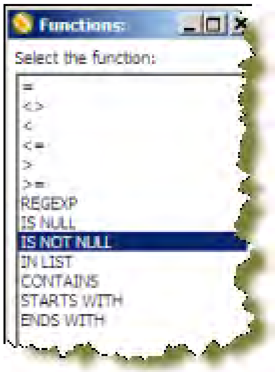
<value>

Fields:

Select a field:

ORDERNUMBER (Integer)
QUANTITYORDERED (Integer)
PRICEEACH (Number)
ORDERLINENUMBER (Integer)
SALES (Number)
ORDERDATE (Date)
STATUS (String)
QTR_ID (Integer)
MONTH_ID (Integer)
YEAR_ID (Integer)
PRODUCTLINE (String)
MSRP (Integer)
PRODUCTCODE (String)
CUSTOMERNAME (String)
PHONE (String)
ADDRESSLINE1 (String)
ADDRESSLINE2 (String)
CITY (String)
STATE (String)
POSTALCODE (String)
COUNTRY (String)
TERRITORY (String)
CONTACTLASTNAME (String)
CONTACTFIRSTNAME (String)

6. In the **Fields:** dialog box select **POSTALCODE** and click **OK**.
7. Click on the comparison operator (set to **=** by default) and select the **IS NOT NULL** function and click **OK**.



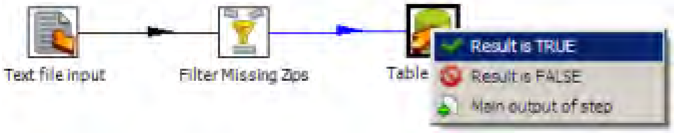
Note: You will return to this step later and configure the **Send true data to step** and **Send false data to step** settings after adding their target steps to your transformation.

8. Save your transformation.

Loading Your Data into a Relational Database

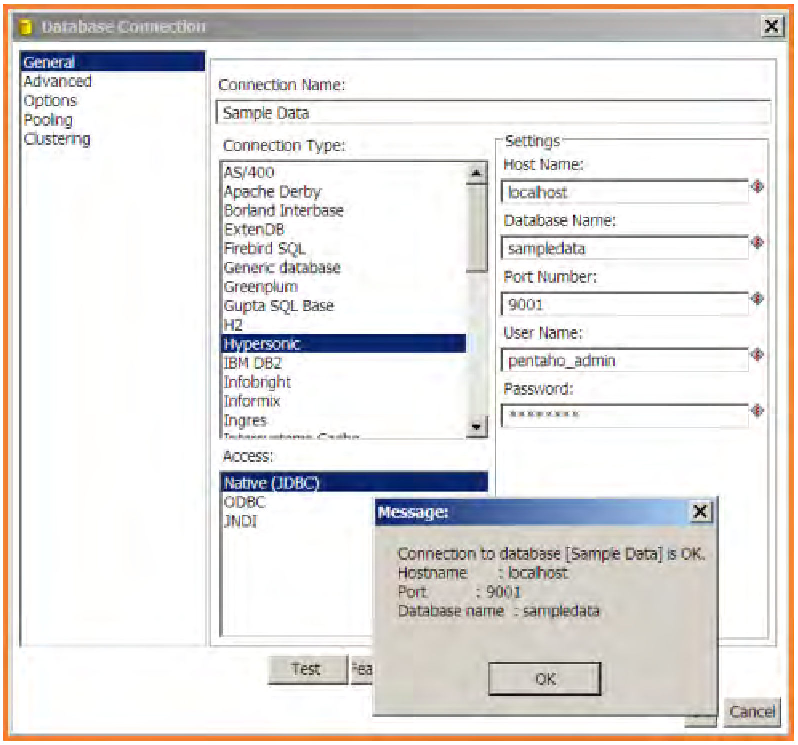
In this exercise you will take all records coming out of our Filter rows step where the POSTALCODE was not null (the **true** condition) and load them into a database table.

1. Under the Design tab, expand the contents of the **Output** folder.
2. Click and drag a **Table Output** step into your transformation; create a hop between the **Filter Missing Zips** (Filter Rows) and **Table Output** steps. Select **Result is TRUE**.



3. Double-click the **Table Output** step to open its edit properties dialog box.

4. Rename your Table Output Step to **Write to Database**.
5. Click **New** next to the **Connection** field. You must create a connection to the database.
   The **Database Connection** dialog box appears.



Connection Name: Sample Data

Host Name: localhost

Database Name: sampledata

Port Number: 9001

User Name: pentaho_admin

Password: ********

Message:

```
Connection to database [Sample Data] is OK.
Hostname      : localhost
Port          : 9001
Database name  : sampledata
```

6. Provide the settings for connecting to the database as shown in the table below.

Connection Name	Type, **Sample Data**
Connection Type:	Choose, Hypersonic
Host Name	localhost
Database Name	Type **sampledata**
Port Number	9001 (keep the default)
User Name	**pentaho_admin**
Password	**password**

7. Click **Test** to make sure your entries are correct. A success message appears. Click **OK**.

   Note: If you get an error testing your connection, ensure that you have provided the correct settings information as described in the talbe and that the sample database is running. See *Starting the Pentaho Data Integration Servers* on page 11 for information about how to start the Data Integration Servers.

8. Click **OK**, to exit the Database Connections dialog box.
9. In the **Table Output** edit properties dialog box, type **SALES_DATA** in the **Target Table** text field.

   This table does not exist in the target database. In the next steps you will generate the Data Definition Language (DDL) to create the table and execute it. DDL are the SQL commands that define the different structures in a database such as CREATE TABLE.
10. Enable the **Truncate Table** property.
11. Click **SQL**generate the DDL for creating our target table.
12. Click **Execute** to run the SQL.
    A results dialog box apperas indicating that one SQL statement was executed. Click **OK** close the execution dialog box. Click **Close** to close the Simple SQL editor dialog box. Click **Close** to close the Table Output edit properties dialog box.

13. Save your transformation.

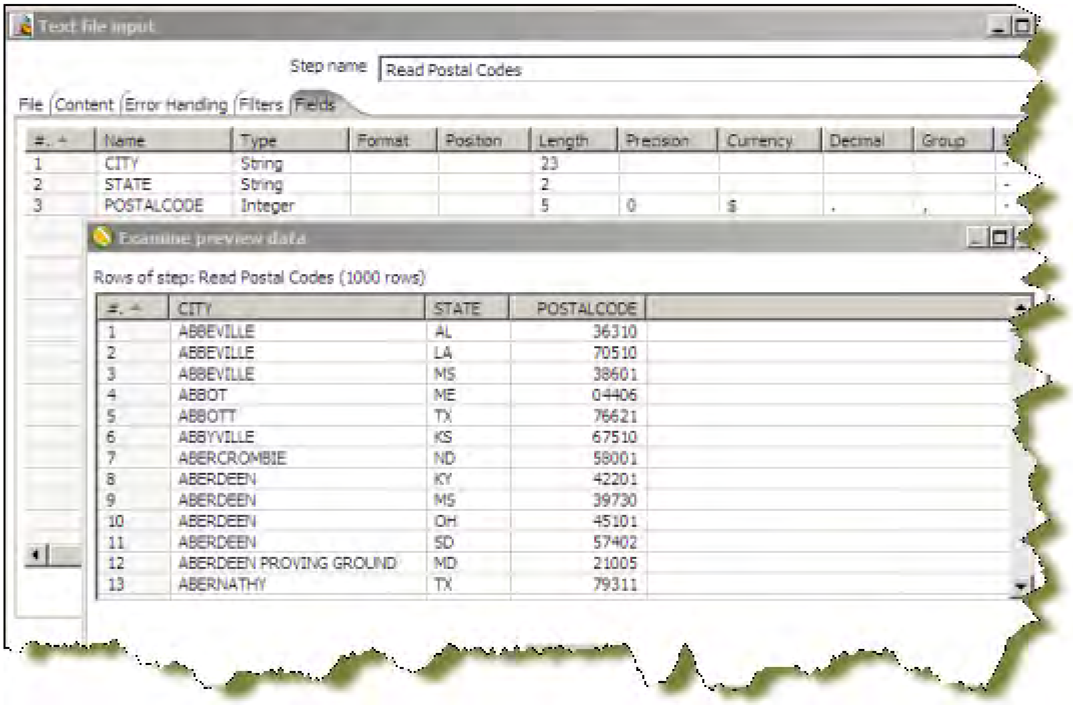
Retrieving Data from your Lookup File

You have been provided a second text file containing a list of cities, states, and postal codes that you will now use to look up the postal codes for all of the records where they were missing (the ‘false’ branch of our Filter rows step). First, you will use a Text file input step to read from the source file, then you will use a Stream lookup step to bring the resolved Postal Codes into the stream.

1. Add a new **Text File input** step to your transformation. In this step you will retrieve the records from your lookup file.
2. Rename your **Text File input** step to, **Read Postal Codes**.
3. Click **Browse** to locate the source file, **Zipssortedbycitystate.csv**, located at `...\pdi-ee\data-integration\samples\transformations\files`.
4. Click **Add**.
   The path to the file appears under **Selected Files**.

   Note: Click **Show File Content** to view the contents of the file. This file is comma (,) delimited, with an enclosure of quotation mark (“), and contains a single header row.

5. Under the **Content** tab, enable the **Header** option. Change the separator character to a comma (,). and confirm that the enclosure setting is correct.
6. Under the **Fields** tab, click **Get Fields** to retrieve the data from your .csv file.
7. Click **Preview Rows** to make sure your entries are correct and click **OK** to exit the Text File input properties dialog box.



Step name: Read Postal Codes

#.	Name	Type	Format	Position	Length	Precision	Currency	Decimal	Group	
1	CITY	String			23					-
2	STATE	String			2					-
3	POSTALCODE	Integer			5	0	$	.	,	-

Rows of step: Read Postal Codes (1000 rows)

#.	CITY	STATE	POSTALCODE
1	ABBEVILLE	AL	36310
2	ABBEVILLE	LA	70510
3	ABBEVILLE	MS	38601
4	ABBOT	ME	04406
5	ABBOTT	TX	76621
6	ABBYVILLE	KS	67510
7	ABERCROMBIE	ND	58001
8	ABERDEEN	KY	42201
9	ABERDEEN	MS	39730
10	ABERDEEN	OH	45101
11	ABERDEEN	SD	57402
12	ABERDEEN PROVING GROUND	MD	21005
13	ABERNATHY	TX	79311

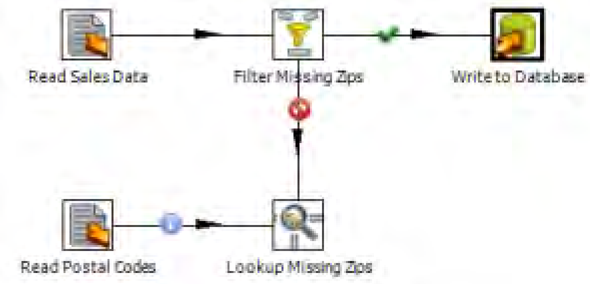
8. Save your transformation.

Resolving Missing Zip Code Information

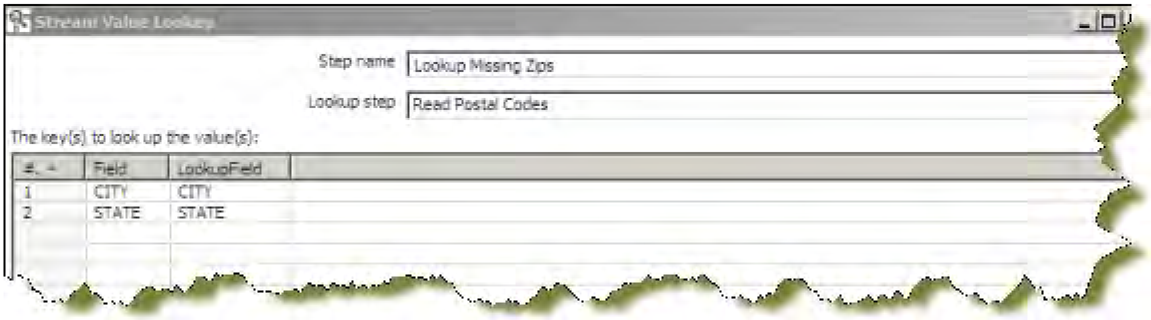
In this exercise, you will begin to resolve the missing zip codes.

1. Add a **Stream Lookup** step to your transformation. Under the Design tab, expand the **Lookup** folder and choose **Stream Lookup**.

2. Draw a hop between the **Filter Missing Zips** (Filter rows) and **Stream Lookup** steps. Select the **Result is FALSE**.
3. Create a hop from the **Read Postal Codes** step (Text File input) to the Stream lookup step.
4. Double-click on the **Stream lookup** step to open its edit properties dialog..
5. Rename Stream Lookup to **Lookup Missing Zips**.



6. Select the **Read Postal Codes** (Text File input) as the **Lookup step**.
7. Define the **CITY** and **STATE** fields in the **key(s) to look up the value(s)** table. Click the drop down in the Field column and selecting **CITY** and then click in the **LookupField** column and select **CITY**. Perform the same actions to define the second key based on the **STATE** fields coming in on the source and lookup streams:



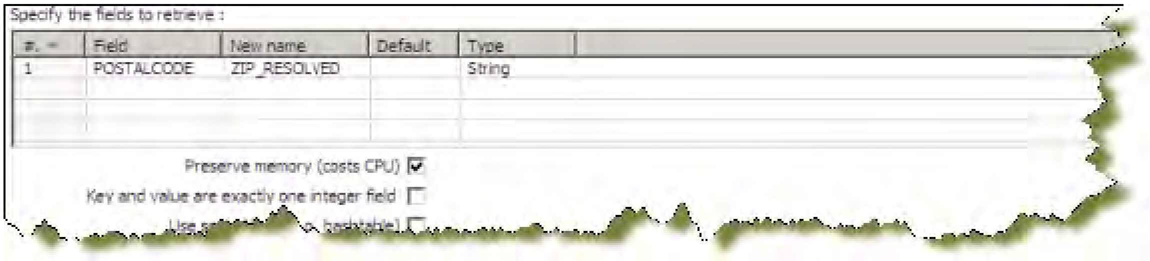
Step name: Lookup Missing Zips

Lookup step: Read Postal Codes

The key(s) to look up the value(s):

#.	Field	LookupField
1	CITY	CITY
2	STATE	STATE

8. Click **Get Lookup Fields**. **POSTALCODE** is the field you want to retrieve. Give it a new name of **ZIP_RESOLVED** and makes sure the **Type** is set to **String**. Click **OK** to close the **Stream Lookup** edit properties dialog box.



Specify the fields to retrieve :

#.	Field	New name	Default	Type
1	POSTALCODE	ZIP_RESOLVED		String

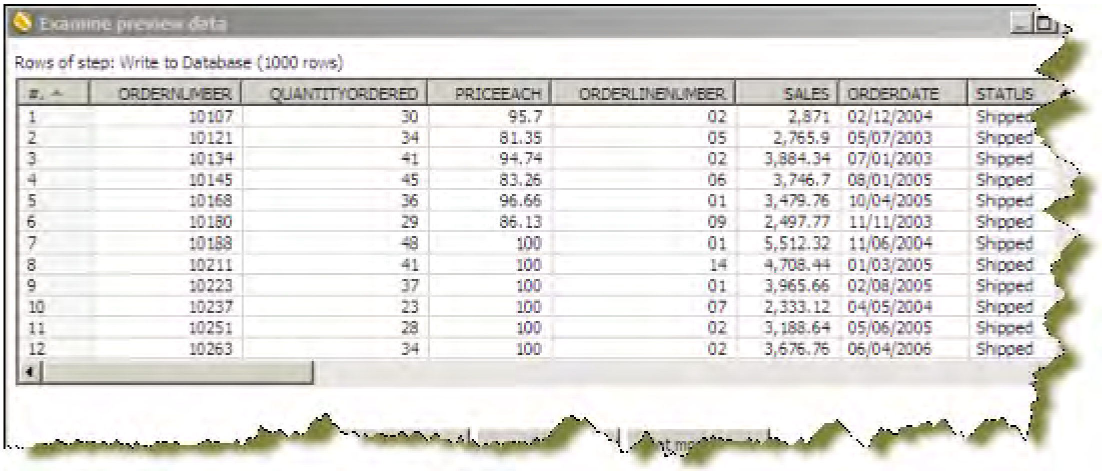
Preserve memory (costs CPU) ☑

Key and value are exactly one integer field □

Use s[illegible] o. hashtable) □

9. Save your transformation.

   You can now select the Lookup Missing Zips step (Stream lookup) in the graphical workspace. Right-click and select **Preview** to display the preview/debugger dialog box. Click **Quick Launch** to preview the data flowing through this step. Notice the new field ZIP_RESOLVED has been added to the stream containing our resolved postal codes.



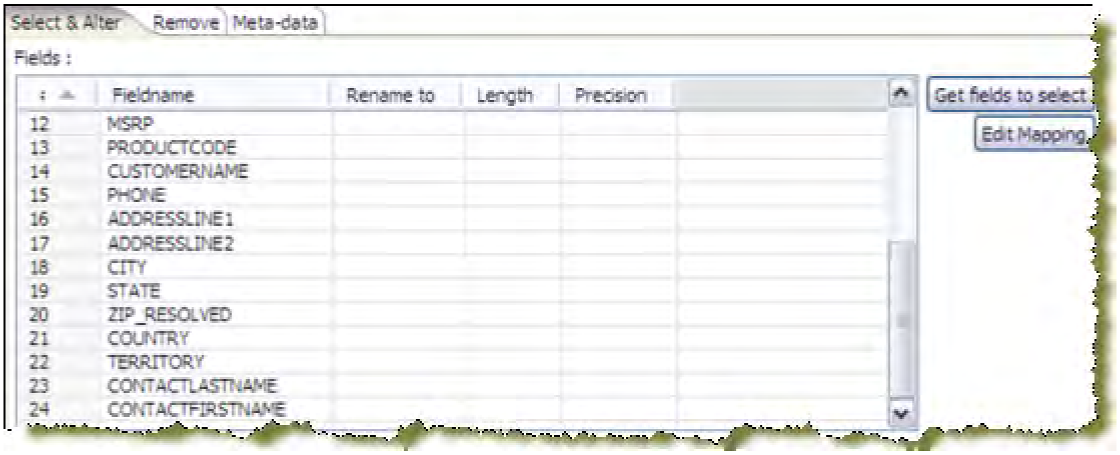
Rows of step: Write to Database (1000 rows)

#.	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS
1	10107	30	95.7	02	2,871	02/12/2004	Shipped
2	10121	34	81.35	05	2,765.9	05/07/2003	Shipped
3	10134	41	94.74	02	3,884.34	07/01/2003	Shipped
4	10145	45	83.26	06	3,746.7	08/01/2005	Shipped
5	10168	36	96.66	01	3,479.76	10/04/2005	Shipped
6	10180	29	86.13	09	2,497.77	11/11/2003	Shipped
7	10188	48	100	01	5,512.32	11/06/2004	Shipped
8	10211	41	100	14	4,708.44	01/03/2005	Shipped
9	10223	37	100	01	3,965.66	02/08/2005	Shipped
10	10237	23	100	07	2,333.12	04/05/2004	Shipped
11	10251	28	100	02	3,188.64	05/06/2005	Shipped
12	10263	34	100	02	3,676.76	06/04/2006	Shipped

Completing your Transformation

The last task is to clean up the field layout on our lookup stream so that it matches the format and layout of our other stream going to the Write to Database (Table output) step. You will create a Select values step. This is a very useful step for renaming fields on the stream, removing unnecessary fields, and more.

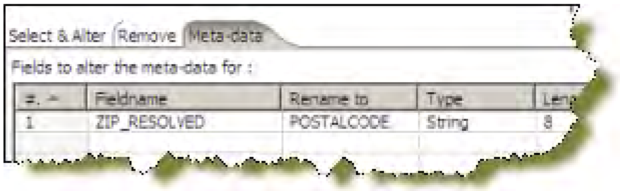
1. Add a **Select Values** step to your transformation. Expand the **Transform** folder and choose **Select Values**.
2. Create a hop between the **Lookup Missing Zips** and **Select Values** steps.
3. Double-click the **Select Values** step to open its properties dialog box.
4. Rename the Select Values step to, **Prepare Field Layout**.
5. Click **Get fields to select** to retrieve all fields and begin modifying the stream layout.
6. Select the **ZIP_RESOLVED** field in the **Fields** list and use <**CTRL**><**UP**> to move it just below the **POSTALCODE** field (the one that still contains null values).
7. Select the old **POSTALCODE** field in the list (line 20) and delete it.



Fields :

#	Fieldname	Rename to	Length	Precision
12	MSRP			
13	PRODUCTCODE			
14	CUSTOMERNAME			
15	PHONE			
16	ADDRESSLINE1			
17	ADDRESSLINE2			
18	CITY			
19	STATE			
20	ZIP_RESOLVED			
21	COUNTRY			
22	TERRITORY			
23	CONTACTLASTNAME			
24	CONTACTFIRSTNAME			

8. The original POSTALCODE field was formatted as an 8-character string. You must modify your new field to match the form. Click the **Meta-Data** tab and click **Get Fields to Change**.
9. In the first row of the **Fields to alter table**, click in the Fieldname column and select ZIP_RESOLVED. **Fields**.
   The output fields match.
10. Type **POSTALCODE** in the **Rename** to column; select **String** in the Type column, and type **8** in the **Length** column. Click **OK** to exit the edit properties dialog box.



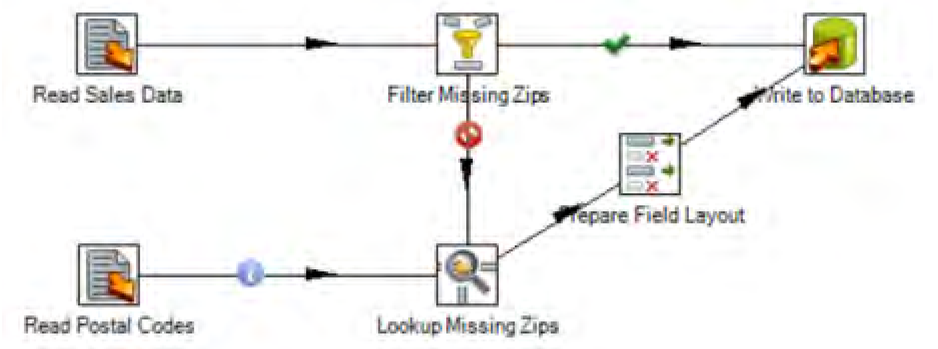
Select & Alter | Remove | Meta-data

Fields to alter the meta-data for :

#.	Fieldname	Rename to	Type	Leng
1	ZIP_RESOLVED	POSTALCODE	String	8

11. Draw a hop from the **Prepare Field Layout** (Select values) step to the **Write to Database** (Table output) step.

12. Save your transformation.



Running Your Transformation

Pentaho Data Integration provides a number of deployment options depending on the needs of your ETL project in terms of performance, batch load window, and so on. The three most common approaches are:

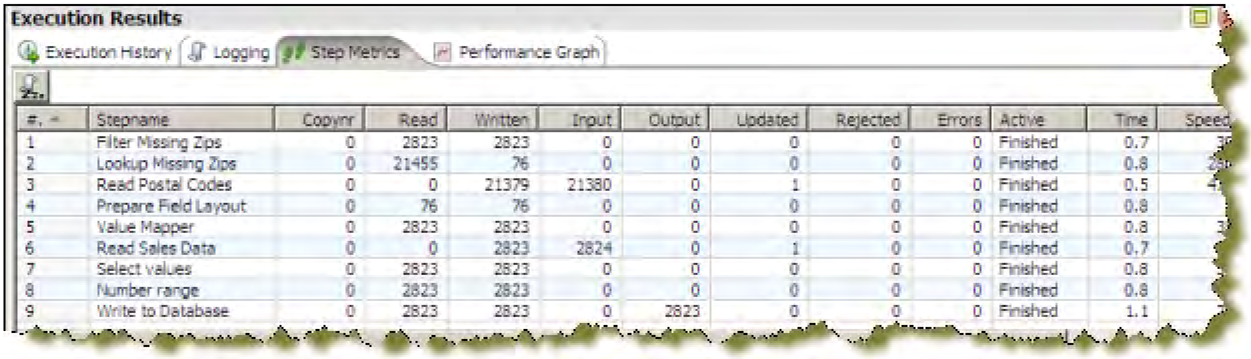
Local execution	Allows you to execute a transformation or job from within the Spoon design environment (on your local machine). This is ideal for designing and testing transformations or lightweight ETL activities
Execute remotely	For more demanding ETL activities, consider setting up a dedicated Enterprise Edition Data Integration Server and using the Execute remotely option in the run dialog. The Enterprise Edition Data Integration Server also enables you to schedule execution in the future or on a recurring basis.
Execute clustered	For even greater scalability or as an option to reduce your execution times, Pentaho Data Integration also supports the notion of clustered execution allowing you to distribute the load across a number of data integration servers.

This final part of the creating a transformation exercise focuses exclusively on the local execution option. For more information on remote, clustered and other execution options review the links in the additional resources section later in this guide or the Pentaho Data Integration User Guide found in the Knowledge Base.

1. In the Spoon graphical interface, click ▶ (Run this Transformation or Job).
The **Execute a Transformation** dialog box appears. You can run a transformation locally, remotely, or in a clustered environment. For the purposes of this exercise, keep the default **Local Execution**.

2. Click **Launch**.

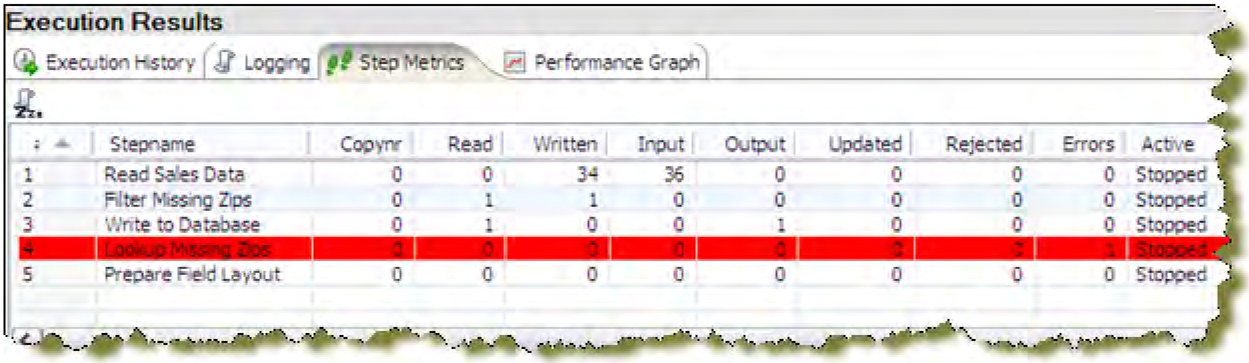
The transformation executes. Upon running the transformation, the **Execution Results** panel opens below the graphical workspace.



Execution Results

#.	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed
1	Filter Missing Zips	0	2823	2823	0	0	0	0	0	Finished	0.7	3
2	Lookup Missing Zips	0	21455	76	0	0	0	0	0	Finished	0.8	26
3	Read Postal Codes	0	0	21379	21380	0	1	0	0	Finished	0.5	4
4	Prepare Field Layout	0	76	76	0	0	0	0	0	Finished	0.8	
5	Value Mapper	0	2823	2823	0	0	0	0	0	Finished	0.8	3
6	Read Sales Data	0	0	2823	2824	0	1	0	0	Finished	0.7	
7	Select values	0	2823	2823	0	0	0	0	0	Finished	0.8	
8	Number range	0	2823	2823	0	0	0	0	0	Finished	0.8	
9	Write to Database	0	2823	2823	0	2823	0	0	0	Finished	1.1	

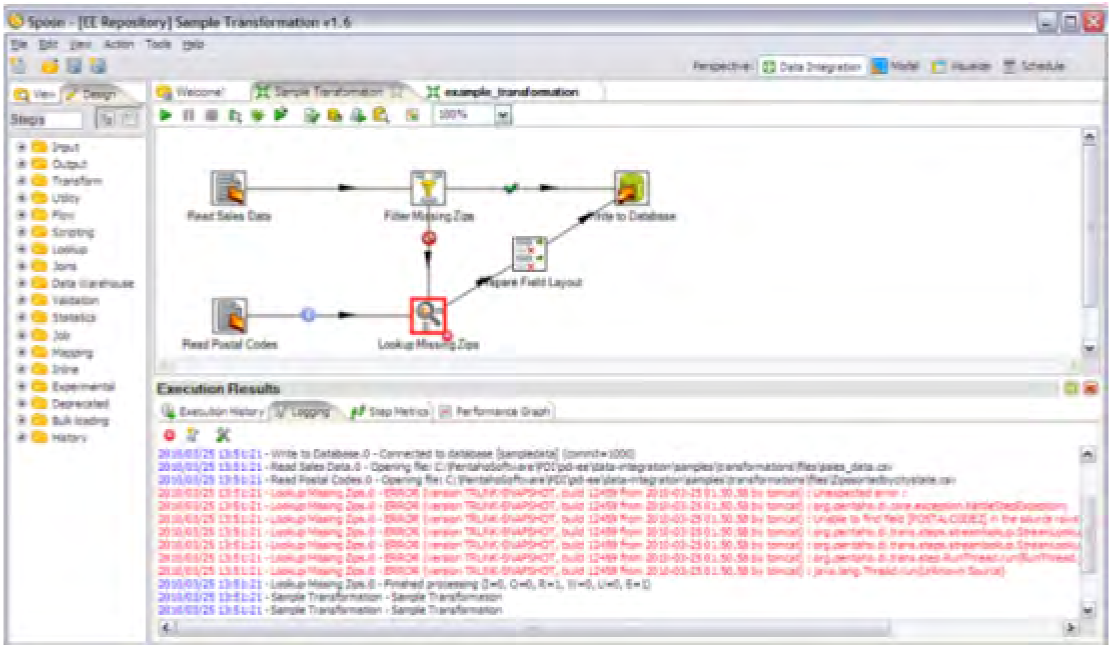
The **Step Metrics** tab provides statistics for each step in your transformation including how many records were read, written, caused an error, processing speed (rows per second) and more. If any of the steps caused the transformation to fail, they would be highlighted in red as shown below.



Execution Results

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active
1	Read Sales Data	0	0	34	36	0	0	0	0	Stopped
2	Filter Missing Zips	0	1	1	0	0	0	0	0	Stopped
3	Write to Database	0	1	0	0	1	0	0	0	Stopped
4	Lookup Missing Zips	0	0	0	0	0	0	0	1	Stopped
5	Prepare Field Layout	0	0	0	0	0	0	0	0	Stopped

The **Logging** tab displays the logging details for the most recent execution of the transformation. Error lines are highlighted in red..

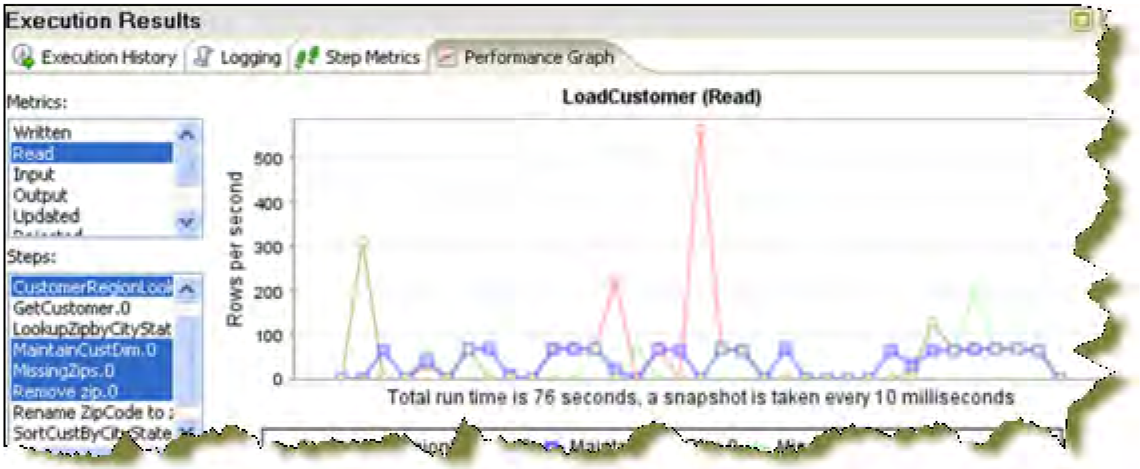


```
2010/03/25 13:51:21 - Write to Database.0 - Connected to database [sampledata] (commit=1000)
2010/03/25 13:51:21 - Read Sales Data.0 - Opening file: C:\PentahoSoftware\PDI\pdi-ee\data-integration\samples\transformations\files\sales_data.csv
2010/03/25 13:51:21 - Read Postal Codes.0 - Opening file: C:\PentahoSoftware\PDI\pdi-ee\data-integration\samples\transformations\files\Zipssortedbycitystate.csv
2010/03/25 13:51:21 - Lookup Missing Zips.0 - ERROR (version TRUNK-SNAPSHOT, build 12459 from 2010-03-25 01.50.58 by tomcat) : Unexpected error :
2010/03/25 13:51:21 - Lookup Missing Zips.0 - ERROR (version TRUNK-SNAPSHOT, build 12459 from 2010-03-25 01.50.58 by tomcat) : org.pentaho.di.core.exception.KettleStepException:
2010/03/25 13:51:21 - Lookup Missing Zips.0 - ERROR (version TRUNK-SNAPSHOT, build 12459 from 2010-03-25 01.50.58 by tomcat) : Unable to find field [POSTALCODE2] in the source rows
2010/03/25 13:51:21 - Lookup Missing Zips.0 - ERROR (version TRUNK-SNAPSHOT, build 12459 from 2010-03-25 01.50.58 by tomcat) : org.pentaho.di.trans.steps.streamlookup.StreamLookup
2010/03/25 13:51:21 - Lookup Missing Zips.0 - ERROR (version TRUNK-SNAPSHOT, build 12459 from 2010-03-25 01.50.58 by tomcat) : org.pentaho.di.trans.steps.streamlookup.StreamLookup
2010/03/25 13:51:21 - Lookup Missing Zips.0 - ERROR (version TRUNK-SNAPSHOT, build 12459 from 2010-03-25 01.50.58 by tomcat) : org.pentaho.di.trans.step.RunThread.run(RunThread
2010/03/25 13:51:21 - Lookup Missing Zips.0 - ERROR (version TRUNK-SNAPSHOT, build 12459 from 2010-03-25 01.50.58 by tomcat) : java.lang.Thread.run(Unknown Source)
2010/03/25 13:51:21 - Lookup Missing Zips.0 - Finished processing (I=0, O=0, R=1, W=0, U=0, E=1)
2010/03/25 13:51:21 - Sample Transformation - Sample Transformation
2010/03/25 13:51:21 - Sample Transformation - Sample Transformation
```

You can see that in this case the Lookup Missing Zips step caused an error because it attempted to lookup values on a field called POSTALCODE2, which did not exist in the lookup stream.

The **Execution History** tab provides you access to the Step Metrics and log information from previous executions of the transformation. This feature ONLY works if you have configured your transformation to log to a database through the Logging tab of the Transformation Settings dialog. For more information on configuring logging or viewing the execution history, please review the links in the additional resources section later in this guide or the Pentaho Data Integration User Guide found in the Knowledge Base.

The **Performance Graph** allows you to analyze the performance of steps based on a variety of metrics including how many records were read, written, caused an error, processing speed (rows per second) and more.



Like the Execution History, this feature requires you to configure your transformation to log to a database through the Logging tab of the Transformation Settings dialog box. For more information on configuring logging or performance monitoring, review the links in the additional resources section later in this guide or the Pentaho Data Integration User Guide found in the Knowledge Base.

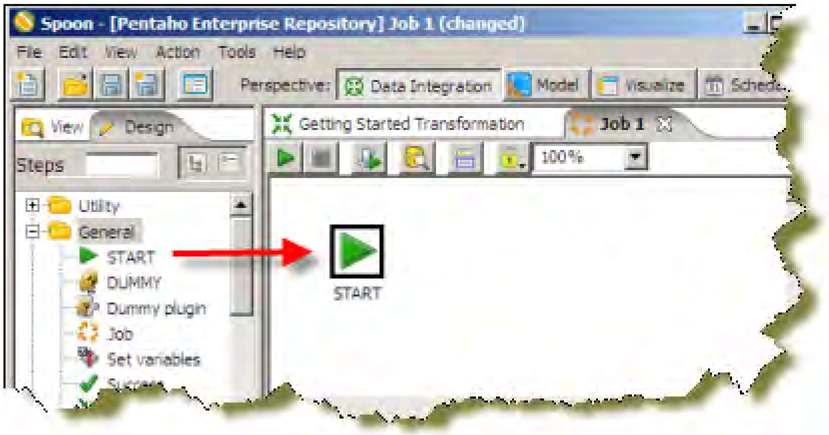
Building Your First Job

Jobs are used to coordinate ETL activities such as:

- Defining the flow and dependencies for what order transformations should be run
- Preparing for execution by checking conditions such as, "Is my source file available or does a table exist?"
- Performing bulk load database operations
- File Management such as posting or retrieving files using FTP, copying files and deleting files
- Sending success or failure notifications through email

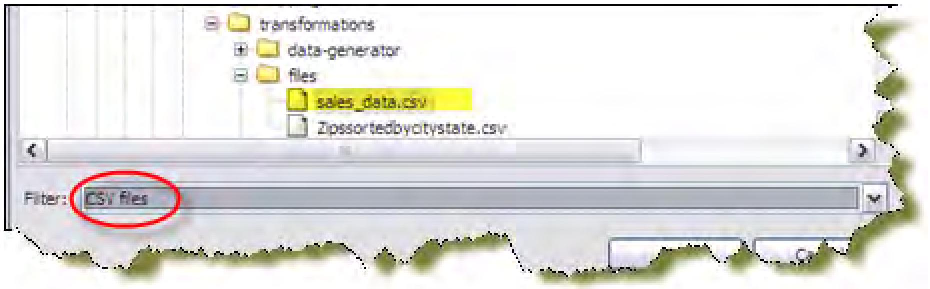
For this exercise, imagine that an external system is responsible for placing your **sales_data.csv** input in its source location every Saturday night at 9 p.m. You want to create a job that will check to see that the file has arrived, run your transformation for loading the records into the database, and send an email if the transformation fails. In a subsequent exercise, you will schedule the job to be run every Sunday morning at 9 a.m.

1. Click (New) in the upper left corner of the Spoon graphical interface.
2. Select **Job** from the list.
3. Expand the **General** folder and drag a **Start** job entry onto the graphical workspace..



   The Start job entry defines where the execution will begin.
4. Expand the **Conditions** folder and add a **File Exists** job entry.
5. Draw a hop from the **Start** job entry to the **File Exists** job entry.
6. Double-click the **File Exists** job entry to open its edit properties dialog box. Click **Browse** and select the **sales_data.csv** from the following location: `...\pdi-ee\data-integration\samples\transformations\files\`.

   Be sure to set the filter to CSV files to see the file.



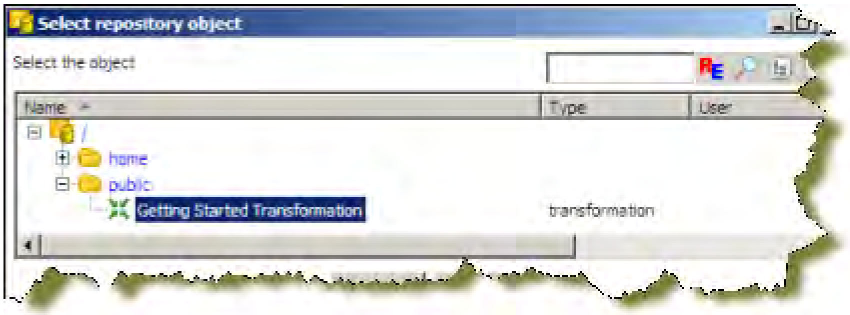
Filter: CSV files

7. Expand the **General** folder and add a **Transformation** job entry.
8. Draw a hop between the **File Exists** and the **Transformation** job entries.

9. Double-click the **Transformation** job entry to open its edit properties dialog box.

10. Select the **Specify by name and directory** option. Click (Browse).

11. Expand the repository tree to find your sample transformation. Select it and click **OK**. You likely have your transformation stored under the "joe," (not public), folder.

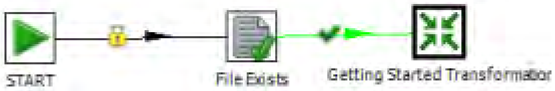


Select repository object

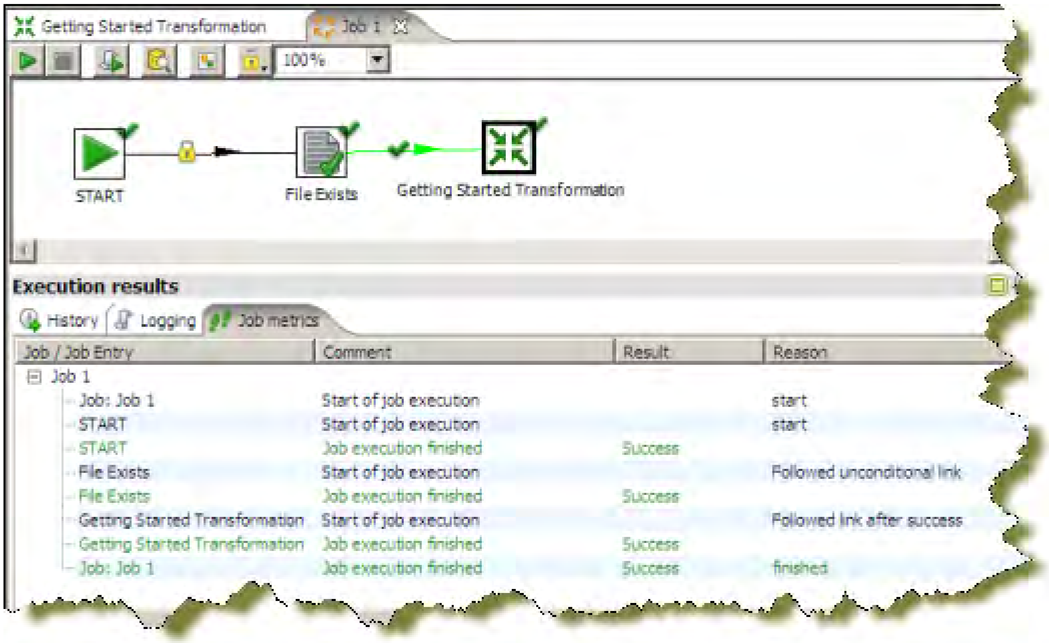
Select the object

Name	Type	User
/		
home		
public		
Getting Started Transformation	transformation	

12. Save your transformation as **Sample Job**.



13. Click (Run Job). When the **Execute a Job** dialog box appears, choose **Local Execution** and click **Launch**.


Getting Started Transformation
Job 1
100%
START
File Exists
Getting Started Transformation

Execution results

History | Logging | Job metrics

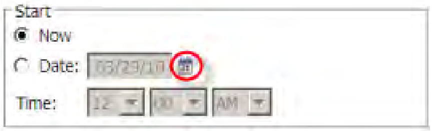
Job / Job Entry	Comment	Result	Reason
Job 1			
Job: Job 1	Start of job execution		start
START	Start of job execution		start
START	Job execution finished	Success	
File Exists	Start of job execution		Followed unconditional link
File Exists	Job execution finished	Success	
Getting Started Transformation	Start of job execution		Followed link after success
Getting Started Transformation	Job execution finished	Success	
Job: Job 1	Job execution finished	Success	finished

The **Execution Results** panel should open showing you the job metrics and log information for the job execution.

Scheduling the Execution of Your Job

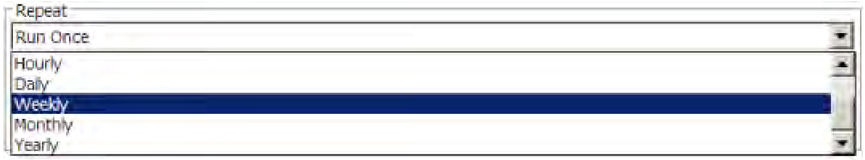
The Enterprise Edition Pentaho Data Integration Server provides scheduling services allowing you to schedule the execution of jobs and transformations in the future or on a recurring basis. In this example, you will create a schedule that runs your Sample Job every Sunday at 9 am..

1. Open your sample job.
2. In the menubar, go to **Action** -> **Schedule**.
   The **Schedule** dialog box appears.
3. For the **Start** option, select the **Date**, click the calendar icon. When the calendar appears, choose the next **Sunday**.

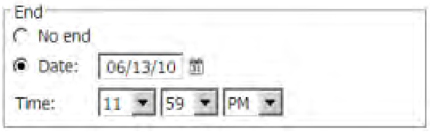
   Start
   ● Now
   ○ Date: 03/29/10
   Time: 12 00 AM

4. Under the **Repeat** section, select the **Weekly** option. Enable the **Sunday** check box.

   Repeat
   Run Once
   Hourly
   Daily
   Weekly
   Monthly
   Yearly

5. For the **End** date, select **Date** and then enter a data several weeks in the future using the calendar picker.

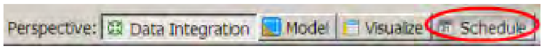
   

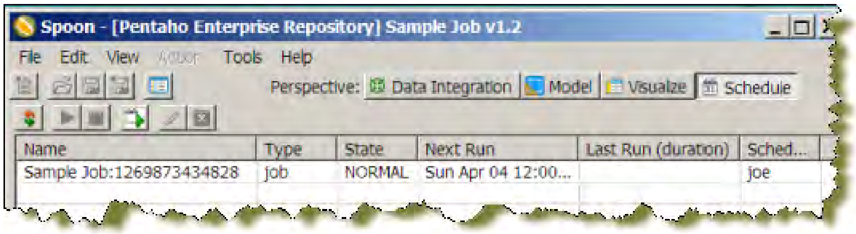
   End
   ○ No end
   ● Date: 06/13/10
   Time: 11 59 PM

6. Click **OK** to complete your schedule.

   Note: The scheduler includes full support for Pentaho Data Integrations parameters, arguments, and variables. For more detailed information on scheduling options, please refer to the Pentaho Data Integration User Guide found in the Knowledge Base (document not available until GA release of Pentaho Data Integration).

7. To view, edit and manage all scheduled activities, click the **Schedule** perspective on the main toolbar. Here you can view a list of all schedules along with information such as when the next scheduled run will take place, when the last run took place and its duration and who scheduled the activity.

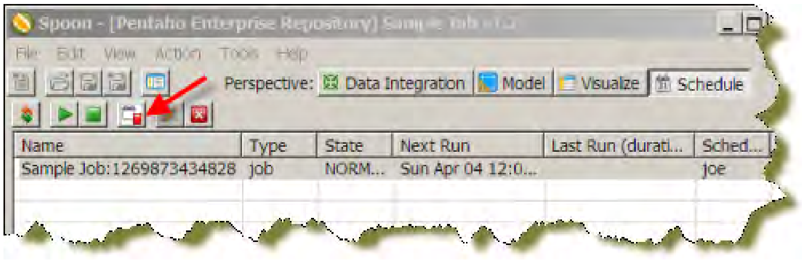
   

   Spoon - [Pentaho Enterprise Repository] Sample Job v1.2

Name	Type	State	Next Run	Last Run (duration)	Sched...
Sample Job:1269873434828	job	NORMAL	Sun Apr 04 12:00...		joe

8. If the scheduler is stopped, you must click (Start Scheduler) on the sub-toolbar. If the button appears with a red stop icon, the scheduler is already running. Your scheduled activity will take place as indicated at the **Next Run** time.



Name	Type	State	Next Run	Last Run (durati...	Sched...
Sample Job:1269873434828	job	NORM...	Sun Apr 04 12:0...		joe

Note: You can also start and stop individual schedules by selecting them in the table and using the Start and Stop buttons ▶■ on the sub-toolbar.

Building Business Intelligence Solutions Using Agile BI

Historically, starting new Business Intelligence projects required careful consideration of a broad set of factors including:

Data Considerations

- Where is my data coming from?
- Where will it be stored?
- What cleansing and enrichment is necessary to address the business needs?

Information Delivery Consideration

- Will information be delivered through static content like pre-canned reports and dashboards?
- Will users need the ability to build their own reports or perform interactive analysis on the data?

Skill Set Considerations

- If users need self-service reporting and analysis, what skill sets do you expect them to have?
- Assuming the project involves some combination of ETL, content creation for reports and dashboards, and meta-data modeling to enable business users to create their own content, do we have all the tools and skill sets to build the solution in a timely fashion?

Cost

- How many tools and from how many vendors will it take to implement the total solution?
- If expanding the use of a BI tool already in house, what are the additional licensing costs associated with rolling it out to a new user community?
- What are the costs in both time and money to train up on all tools necessary to roll out the solution?
- How long is the project going to take and when will we start seeing some ROI?

Because of this, many new projects are scratched before they even begin. Pentaho’s Agile BI initiative seeks to break down the barriers to expanding your use of Business Intelligence through an iterative approach to scoping, prototyping, and building complete BI solutions. It is an approach that centers on the business needs first, empowers the business users to get involved at every phase of development, and prevents projects from going completely off track from the original business goals.

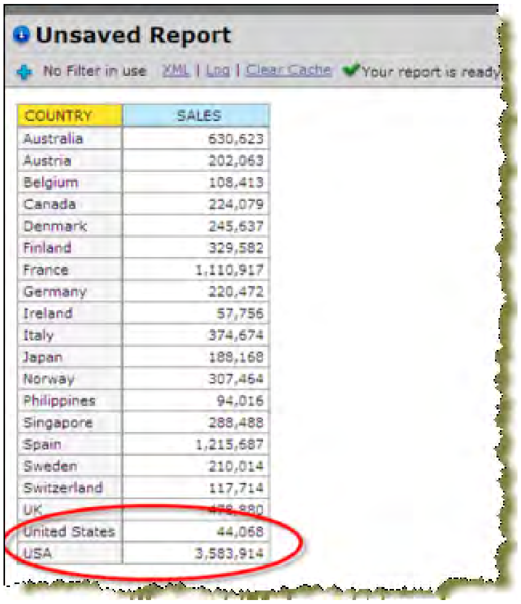
In support of the Agile BI methodology, the Spoon design environment provides an integrated design environment for performing all tasks related to building a BI solution including ETL, reporting and OLAP metadata modeling and end user visualization. In a single click, Business users will instantly be able to start interacting with data, building reports with zero knowledge of SQL or MDX, and work hand in hand with solution architects to refine the solution.

Using Agile BI

This exercise builds upon your sample transformation and highlights the power an integrated design environment can provide for building solutions using Agile BI.

For this example, your business users have asked to see what the top 10 countries are based on sales. Furthermore, they want the data broken down by deal size where small deals are those less than $3,000, medium sized deals are between $3,000 and $7,000, and large deals are over $7,000.

1. Open or select that tab for the sample transformation created earlier in this guide.
2. Right-click the **Write to Database** (Table Output) step, and select **Visualize** -> **Analyzer**.
   In the background, Pentaho Data Integration automatically generates the OLAP model that allows you to begin interacting immediately with your new data source.
3. Drag the **COUNTRY** field from the **Field** list on the left onto the report.
4. Drag the **SALES** measure from the **Field** list onto the report.



Unsaved Report

No Filter in use XML | Log | Clear Cache ✓Your report is ready

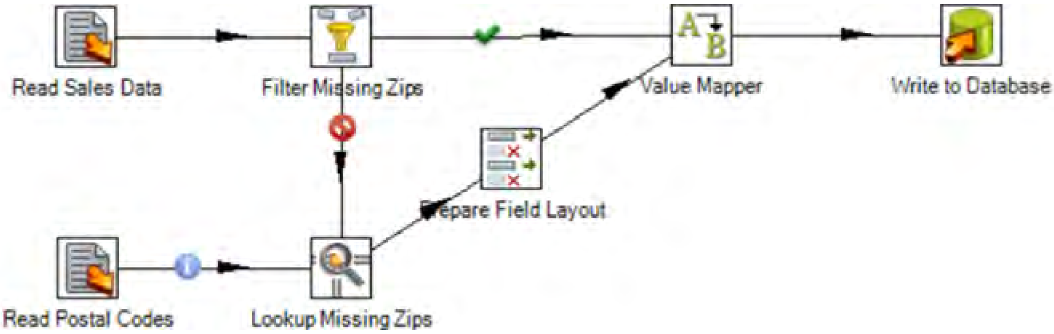
COUNTRY	SALES
Australia	630,623
Austria	202,063
Belgium	108,413
Canada	224,079
Denmark	245,637
Finland	329,582
France	1,110,917
Germany	220,472
Ireland	57,756
Italy	374,674
Japan	188,168
Norway	307,464
Philippines	94,016
Singapore	288,488
Spain	1,215,687
Sweden	210,014
Switzerland	117,714
UK	478,880
United States	44,068
USA	3,583,914

Note: Immediately you can see that there is another problem with the quality of the data. You can see that some records being loaded into the database have a COUNTRY value of *United States*, while others have a value of *USA*. In the next steps, you will return to the data integration perspective to resolve this issue.

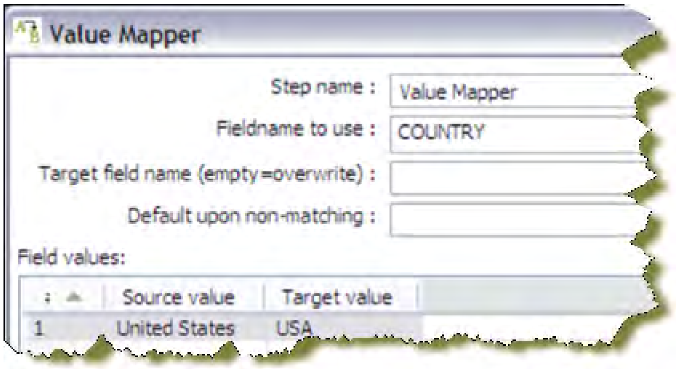
Correcting the Data Quality Issue

Follow the instructions below to correct the data quality issue:

1. Click on the **Data Integration** perspective in the main toolbar.
2. Right-click the **Table output** step from the flow and choose **Detach step**. Repeat this process to detach the second hop.
3. Expand the **Transform** folder in the Design Palette and add a **Value Mapper** step to the transformation.
4. Draw a hop from the **Filter Missing Zips** (Filter rows) step to the **Value Mapper** step and select **Result is TRUE**.
5. Draw a hop from the **Prepare Field Layout** (Select values) step to the **Value Mapper** step.
6. Draw a hop from the **Value Mapper** step to the **Write to Database** (Table output) step. Your transformation should look like the sample below:



7. Double-click on the **Value Mapper** step to open its edit step properties dialog box.
8. Select the **COUNTRY** field in the **Fieldname** to use input.
9. In the first row of the **Field Values** table, type **United States** as the **Source** value and **USA** as the **Target value**. Click **OK** to exit the dialog box.



Value Mapper

Step name : Value Mapper

Fieldname to use : COUNTRY

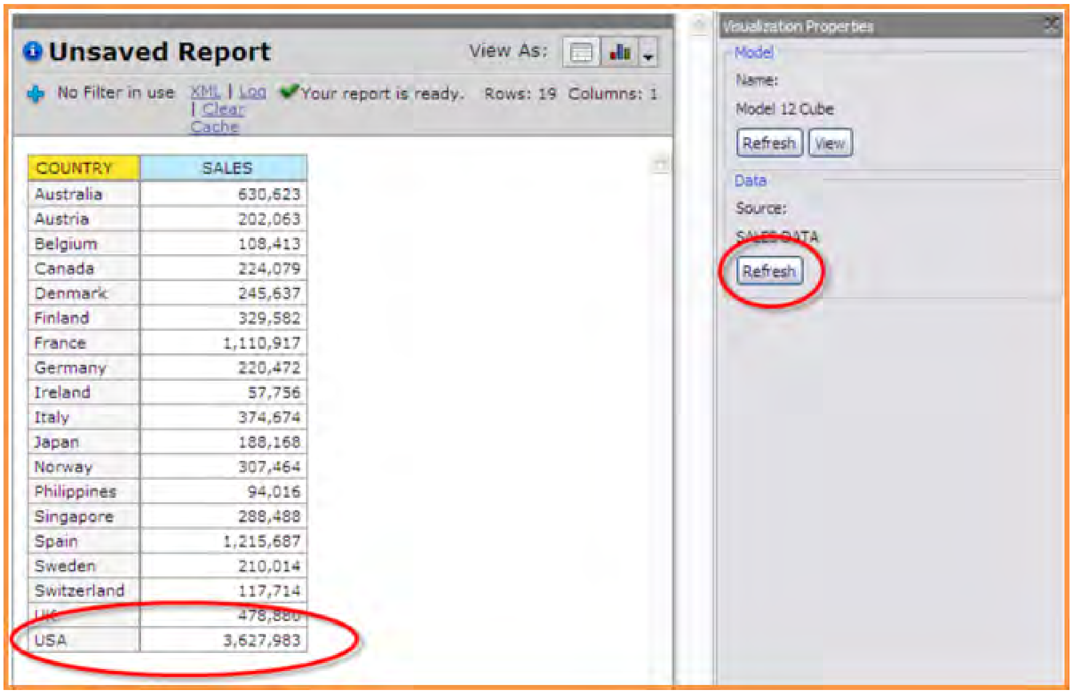
Target field name (empty=overwrite) :

Default upon non-matching :

Field values:

#	Source value	Target value
1	United States	USA

10. Save and run the transformation.
11. Click **Visualize** in the main toolbar.
12. Click the **Clear Cache** link at the top of the report.
13. Click **Refresh** under the data section of the **Visualization Properties** panel and notice that the data has been cleansed.



Unsaved Report

View As:

No Filter in use XML | Log | Clear Cache ✓Your report is ready. Rows: 19 Columns: 1

COUNTRY	SALES
Australia	630,623
Austria	202,063
Belgium	108,413
Canada	224,079
Denmark	245,637
Finland	329,582
France	1,110,917
Germany	220,472
Ireland	57,756
Italy	374,674
Japan	188,168
Norway	307,464
Philippines	94,016
Singapore	288,488
Spain	1,215,687
Sweden	210,014
Switzerland	117,714
UK	478,880
USA	3,627,983

Visualization Properties

Model

Name:

Model 12 Cube

Refresh View

Data

Source:

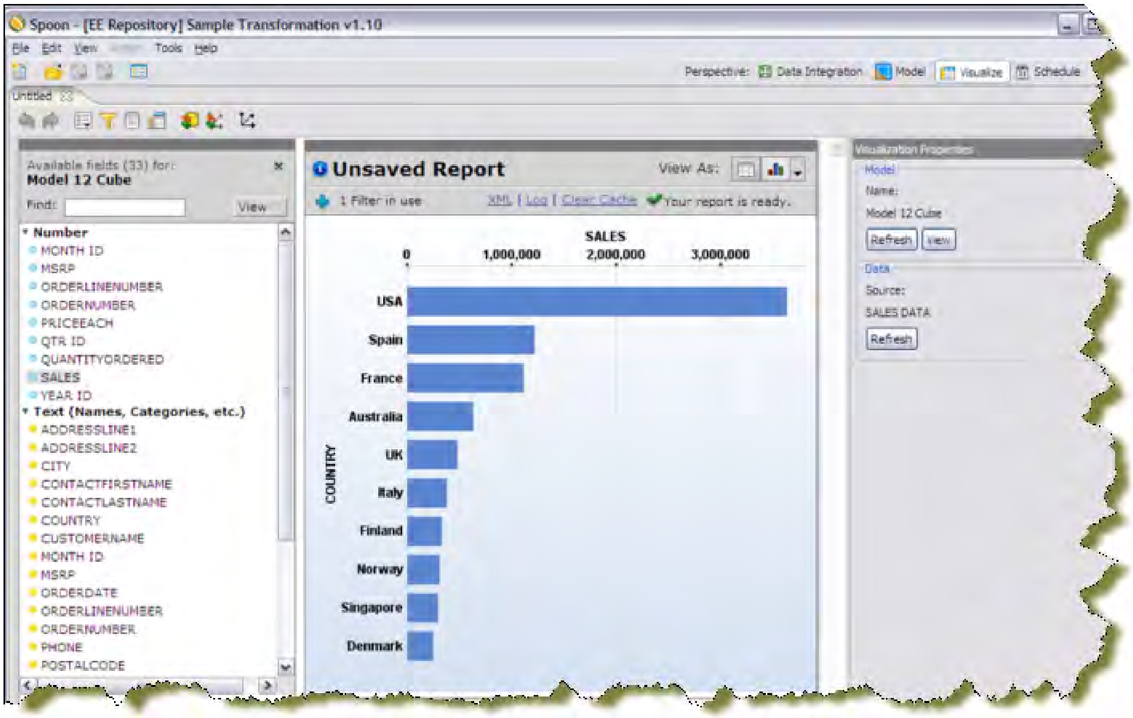
SALES DATA

Refresh

Creating a Top Ten Countries by Sales Chart

Follow the instructions below to create the top ten countries by sales chart:

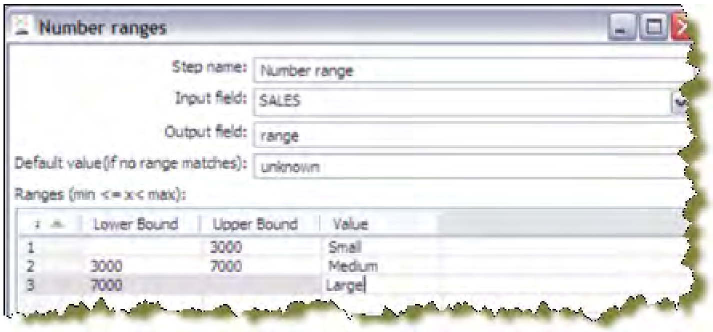
1. Right-click the **COUNTRY** header and select **Top 10**, and so on..
2. Confirm that the default settings are set to return the Top 10 **COUNTRY** members by the **SALES** measure. Click **OK**.
3. Click 📊 (chart) and select **Bar** to change the visualization to a bar chart.



Breaking Down Your Chart by Deal Size

Your source data does not contain an attribute for Deal Size, so you will use the Data Integration perspective to add the new field.

1. Click **Data Integration** in the main toolbar..
2. Expand the **Transform** folder and drag a **Number range** step onto the graphical workspace.
3. Drag the **Select Values** step onto the hop connecting the **Value Mapper** and **Write to Database** (Table output) steps. Choose **Yes** to split the hop.
4. Double-click the **Select Values** step to open its edit step properties dialog box.
5. Double-click the **Select Values** step to open its editor properties dialog box.
6. Click the **Metadata** tab.
7. Select the **SALES** field as your **Input** field.
8. Type **Deal Size** as the name for the **Output** field.
9. In the **Fields to alter** table, select the **SALES** field in as the **Fieldname**, select **Number** as the Type, and Enter **#.#** as the **Format**. Click **OK**.
10. Expand the **Transform** folder and drag a **Number range** step onto graphical workspace.
11. Drag the **Number range** step onto the hop connecting the **Select Values** and **Write to Database** (Table output) steps. Select **Yes** to split the hop.
12. Double-click **Number range** to open its edit properties dialog box.
13. Choose the **SALES** field as your Input field.
14. Type **Deal Size** as the name for the **Output** field.
15. In the **Ranges** table, define number ranges as shown in the example below. Click **OK**.



Number ranges

Step name: Number range

Input field: SALES

Output field: range

Default value(if no range matches): unknown

Ranges (min <= x < max):

#	Lower Bound	Upper Bound	Value
1		3000	Small
2	3000	7000	Medium
3	7000		Large

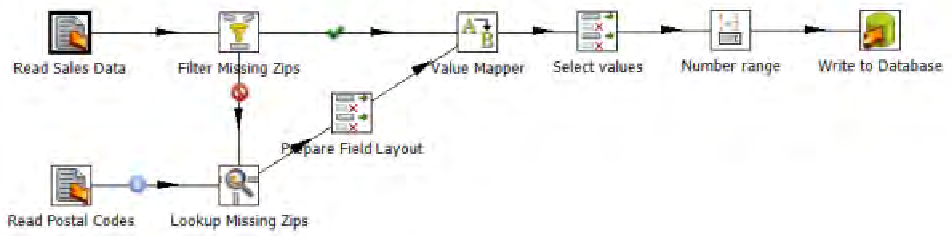
Note: Because this step will be adding new field onto the stream, you must update your target database table to add the new column in the next steps.

16. Double-click on the **Write to Database** (Table output) step.

17. Click **SQL** to generate the DDL necessary to update the target table.

18. Click **Execute** to run the SQL. Click **OK** to close the results dialog box. Click **Close** to exit the **Simple SQL Editor** dialog box. Click **OK** to close the edit step properties dialog.

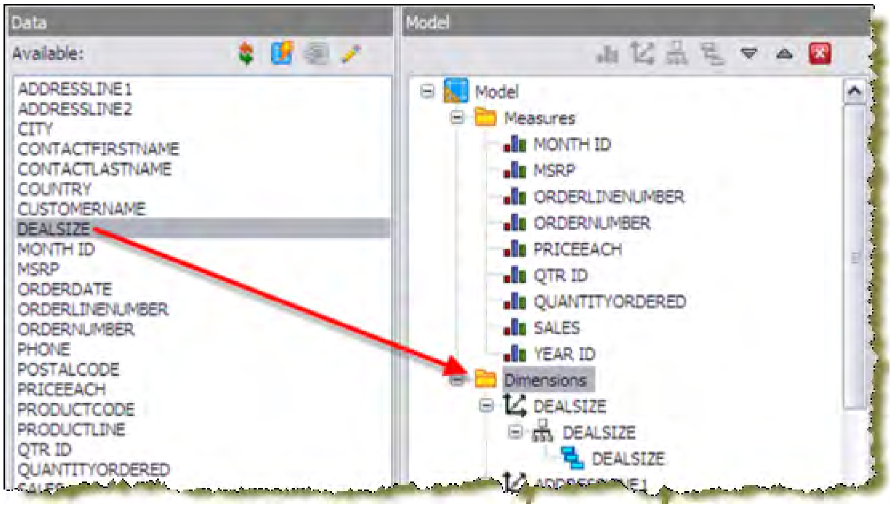
19. Save and run your transformation to re-populate your target database.



Wrapping it Up

Follow the instructions below to complete your Agile BI exercise:

1. Click **Visualize** to return to your **Top 10 Countries** chart. Next, you will update your dimensional model with the new **Deal Size** attribute.
2. Click **View** in the Visualization Properties panel on the right to display the **Model** perspective and begin editing the model used to build your chart.
3. Drag the **DEALSIZE** field from the list of available fields on the left onto the Dimensions folder in the Model panel in the middle. This adds a new dimension called **DEALSIZE** with a single default hierarchy and level of the same name.



Data

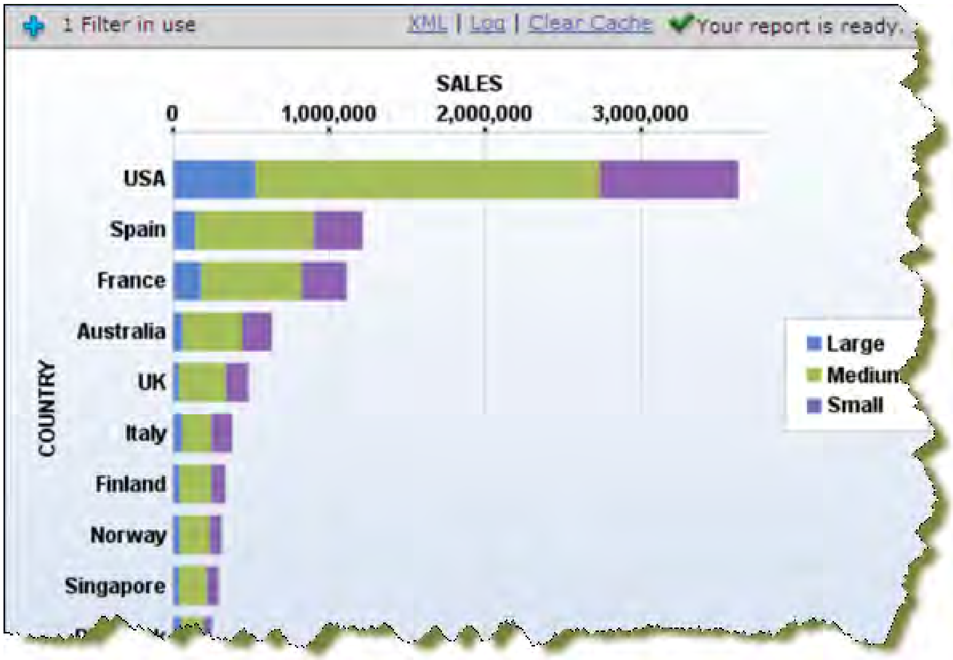
Available:

ADDRESSLINE1
ADDRESSLINE2
CITY
CONTACTFIRSTNAME
CONTACTLASTNAME
COUNTRY
CUSTOMERNAME
DEALSIZE
MONTH ID
MSRP
ORDERDATE
ORDERLINENUMBER
ORDERNUMBER
PHONE
POSTALCODE
PRICEEACH
PRODUCTCODE
PRODUCTLINE
QTR ID
QUANTITYORDERED

Model

- Model
  - Measures
    - MONTH ID
    - MSRP
    - ORDERLINENUMBER
    - ORDERNUMBER
    - PRICEEACH
    - QTR ID
    - QUANTITYORDERED
    - SALES
    - YEAR ID
  - Dimensions
    - DEALSIZE
      - DEALSIZE
        - DEALSIZE

4. Click **Save** on the main toolbar to save your updated model. Click **Visualize** to return to your Top 10 Countries chart.
5. Click **Refresh** to update your field list to include the new **DEALSIZE** attribute.
6. Click **Chart** and change the chart type to **Stacked Bar**.
7. Click (Toggle Layout) to open the **Layout** panel.
8. Drag **DEALSIZE** from the field list on the left into the **Color Stack** section of the **Layout** panel.
9. Click (Toggle Layout) to close the Layout Panel. You have successfully delivered your business user's request



Why Choose Enterprise Edition?

Pentaho Data Integration Enterprise Edition enables you to deploy the world's most popular open source Data Integration solution with confidence, security, and far lower total cost of ownership than proprietary alternatives. Pentaho Data Integration Enterprise Edition provides additional capabilities including a comprehensive professional technical support program, software maintenance releases, enhanced software functionality, certified software, product expertise, and the best software assurance program in the industry.

For more information or to start your subscription today, contact us at *http://www.pentaho.com/contact/*