Get Started with DI

Help and Support Resources

If you do not find answers to your quesions here, please contact your Pentaho technical support representative.

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, visit http://www.pentaho.com/training.

Liability Limits and Warranty Disclaimer

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

Pentaho (TM) and the Pentaho logo are registered trademarks of Pentaho Corporation. All other trademarks are the property of their respective owners. Trademarked names may appear throughout this document. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of the trademarked name, Pentaho states that it is using the names for editorial purposes only and to the benefit of the trademark owner, with no intention of infringing upon that trademark.

Third-Party Open Source Software

For a listing of open source software used by each Pentaho component, navigate to the folder that contains the Pentaho component. Within that folder, locate a folder named licenses. The licenses folder contains HTML.files that list the names of open source software, their licenses, and required attributions.

Contact Us

Global Headquarters 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

Sales Inquiries: sales@pentaho.com

Contents

Introduction

Pentaho Data Integration (PDI) is an extract, transform, and load (ETL) solution that uses an innovative metadata-driven approach.

PDI includes the DI Server, a design tool, three utilities, and several plugins.

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
- Rapid prototyping of ROLAP schemas
- Hadoop functions: Hadoop job execution and scheduling, simple Hadoop MapReduce design, Amazon EMR integration

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

Use Perspectives Within Spoon

Pentaho Data Integration (PDI) provides you with tools that include ETL, modeling, and visualization in one unified environment — the Spoon interface. This integrated environment allows you to work in close cooperation with business users to build business intelligence solutions more quickly and efficiently.
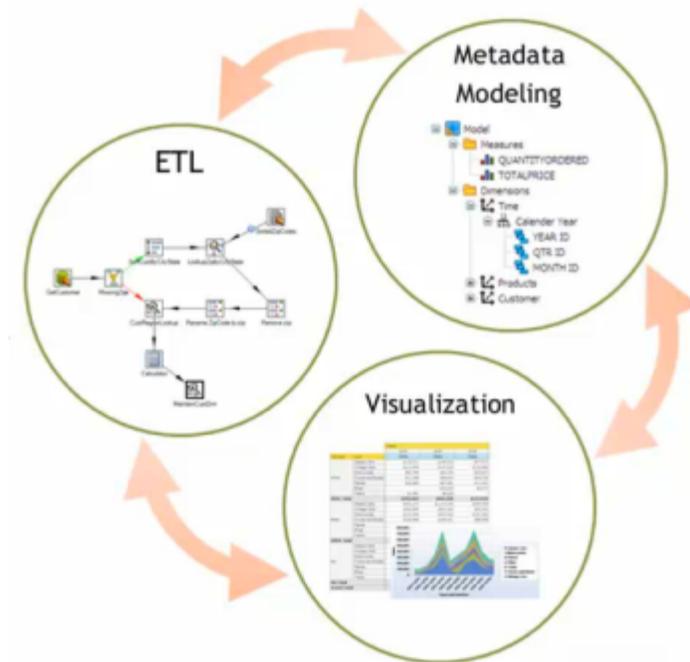
When you are working in Spoon you can *change perspectives*, or switch from designing ETL jobs and transformations to modeling your data, and visualizing it. As users provide you with feedback about how the data is presented to them, you can quickly make iterative changes to your data directly in Spoon by changing perspectives. The ability to quickly respond to feedback and to collaborate with business users is part of the Pentaho Agile BI initiative.

From within Spoon you can change perspectives using the **Perspective** toolbar located in the upper-right corner.
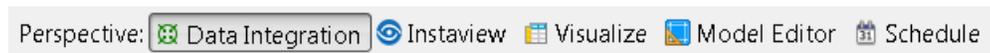

The perspectives in PDI enable you to focus how you work with different aspects of data.

- *Data Integration perspective*—Connect to data sources and extract, transform, and load your data
- *Model perspective*—Create a metadata model to identify the relationships within your data structure
- *Forecast perspective*—Identify trends within facets of your data
- *Visualize perspective*—Create charts, maps, and diagrams based on your data
- *Instaview perspective*—Create a data connection, a metadata model, and analysis reports all at once with a dialog-guided, template-based reporting tool
- **Schedule** perspective—Plan when to run data integration jobs and set timed intervals to automatically send the output to your preferred destinations
- **ScatterPlot3D perspective*—Visualize your data as a Java 3D scatter plot visualization or histogram matrix overview (**separate installation required*)

Create Transformations

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:
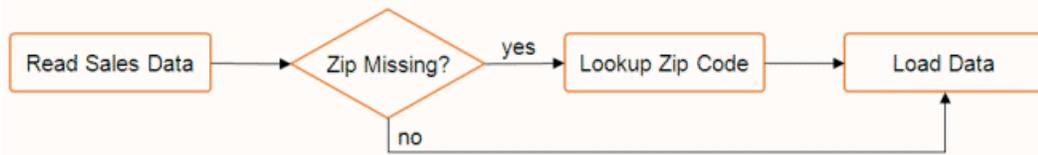
Retrieve Data from a Flat File

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.
3. Under *the Design tab*, expand the **Input** node; then, select and drag a **Text File Input** step onto the *canvas*.
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.
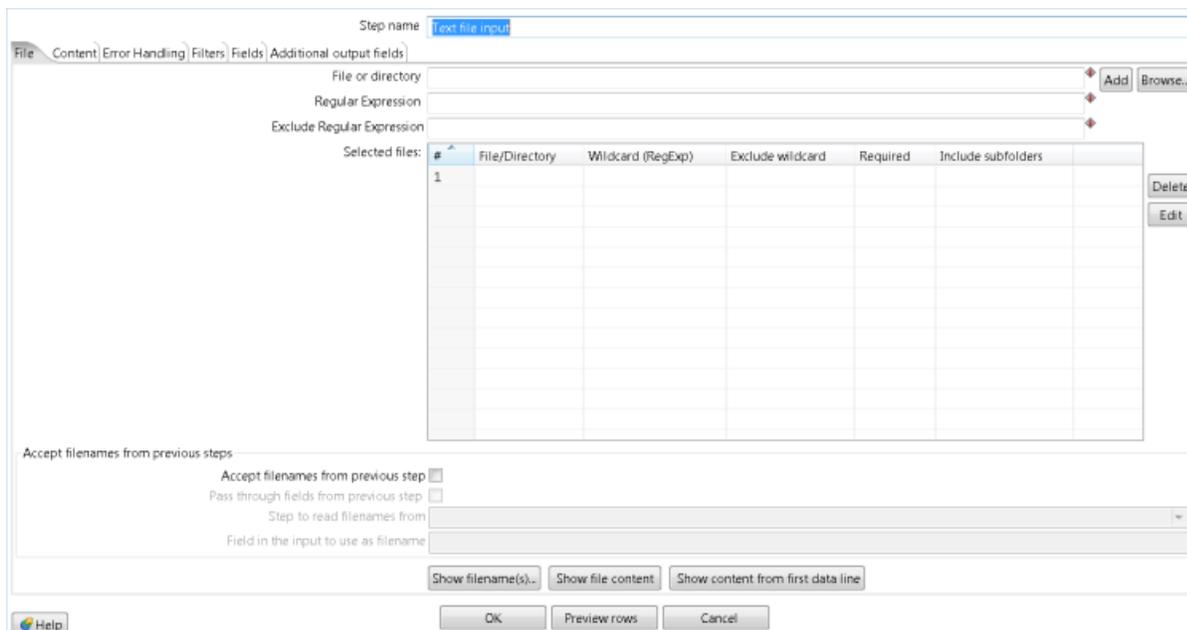
5. In the **Step Name** field, type **Read Sales Data**.

   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 `...\design-tools\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 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.
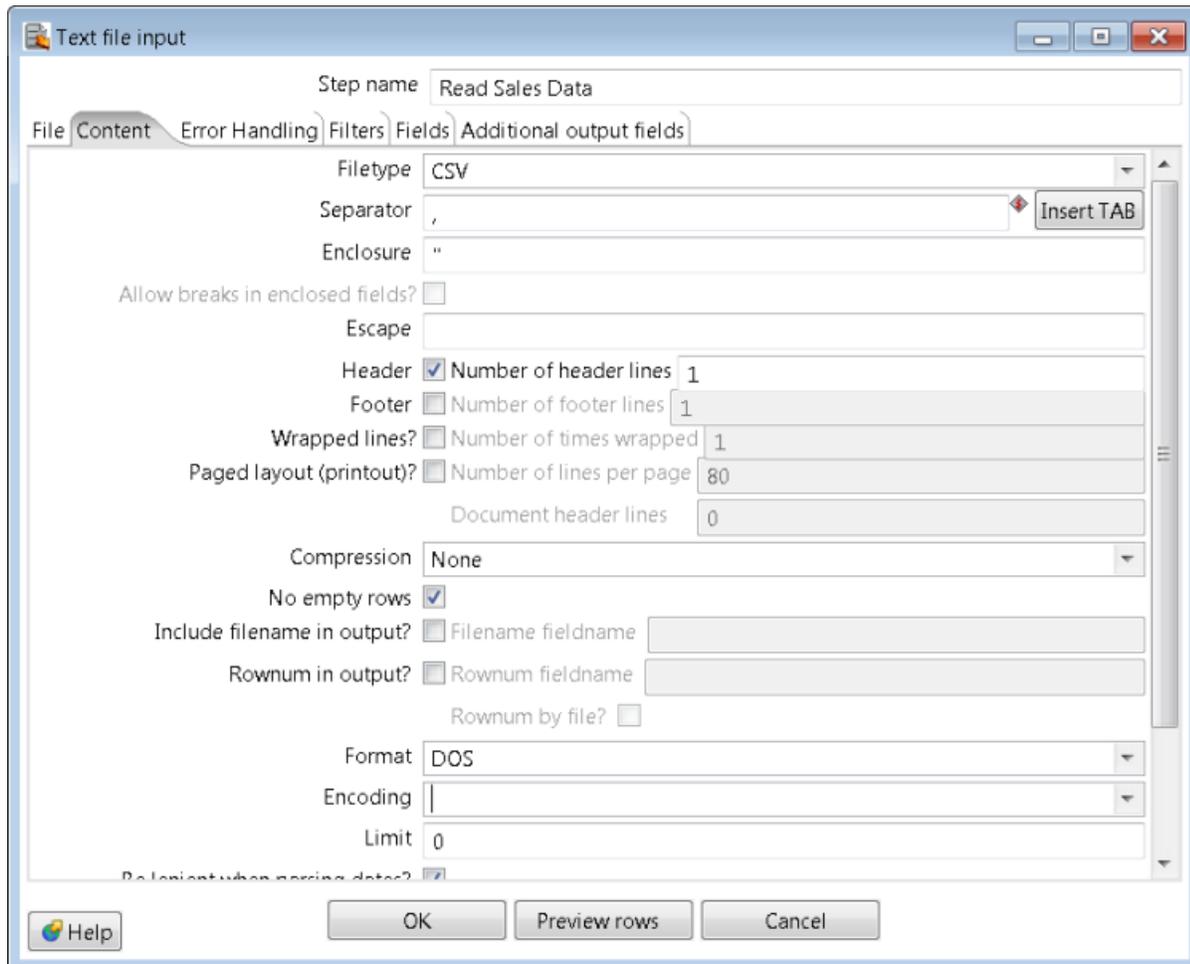
10. Click the **Fields** tab and click **Get Fields** to retrieve the input fields from your source file.

11. Click **Preview Rows** to verify that your file is being read correctly. You can change the number of rows to preview. Click **OK** to exit the step properties dialog box.

12. *Save your transformation*.

Save Your Transformation

Save the transformation you created in *Retrieve Data from a Flat File*. You can save transformations at any time.

1. In the Spoon designer, click **File** > **Save As**.
   The **Transformation Properties** dialog box appears.

2. In the **Transformation Name** field, type **Getting Started Transformation**.

3. In the **Directory** field, click the folder icon.

4. Expand the **Home** directory and double-click the folder in which you want to save..
   Your transformation saves in the folder in the DI repository.

5. Click **OK** to exit the **Transformation Properties** dialog box.
   The **Enter Comment** dialog box appears.

6. Click in the **Enter Comment** dialog box and press <**Delete**> to remove the default text string. Type a meaningful comment about your transformation.
   The comment and your transformation are tracked for version control purposes in the DI repository.

7. Click **OK** to exit the **Enter Comment** dialog box.

Filter Records with Missing Postal Codes

After completing *Retrieve Data from a Flat File*, you are ready to add the next step to your transformation. The source file contains several records that are missing postal codes. Use the Filter Rows transformation step to separate out those records so that you can resolve them in a later exercise.

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.
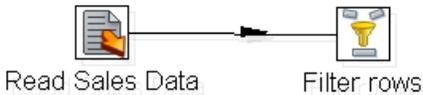
3. Double-click the **Filter Rows** step.
   The **Filter Rows** edit properties dialog box appears.

4. In the **Step Name** field type, **Filter Missing Zips**.

5. Under **The condition**, click `<field>`. The **Fields:** dialog appears. These are the conditions you can select.

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**. Click **OK** to exit the Filter Rows properties dialog box.

   > **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.

Load Your Data into a Relational Database

After completing *Filter Records with Missing Postal Codes*, you are ready to take all records exiting the 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** node.

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. In the dialog that appears, 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.

6. Provide the settings for connecting to the database.

Connection Name	Sample Data
Connection Type:	H2
Host Name	localhost
Database Name	sampledata
Port Number	9092

User Name	sa
Password	blank/no password

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

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

8. Click **OK**, to exit the Database Connections dialog box.

9. 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. In the dialog box, enable the **Truncate Table** property.

11. Click the **SQL** button at the bottom of the **Table output** dialog box to generate the DDL for creating your target table.

12. Click **Execute** to run the SQL.
    The **Simple SQL editor** dialog box appears. Click **Execute** to execute the SQL statement and then click **Close** to close the **Simple SQL editor** dialog box. Click **OK** to close the **Table output** dialog box.

13. Save your transformation.

Retrieve Data from your Lookup File

After you *Load Your Data into a Relational Database*, you are ready to retrieve 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 your 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 `...\design-tools\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, change the separator character to a comma (,). and confirm that the enclosure setting is a quotation mark ("}. Enable the **Header** option is selected.

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.

8. Click **OK** to save your transformation.

Resolve Missing Zip Code Information

After you *Retrieve Data from your Lookup File*, you can 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** and **Stream Lookup** steps. In the dialog box that appears, select **Result is FALSE**.

3. Create a hop from the **Read Postal Codes** step to the Stream lookup step.

4. Double-click on the **Stream lookup** step to open its edit properties dialog box.

5. Rename Stream Lookup to **Lookup Missing Zips**.

6. From the **Lookup step** drop-down box, select **Read Postal Codes** 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 select **CITY**. 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	
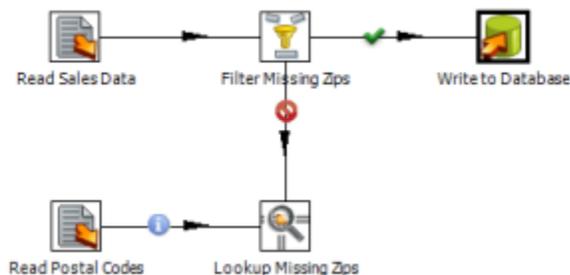
8. Click **Get Lookup Fields**. **POSTALCODE** is the only field you want to retrieve. To delete the **CITY** and **STATE** lines, right-click in the line and select **Delete Selected Line**. Give **POSTALCODE** a new name of **ZIP_RESOLVED** and make sure the **Type** is set to **String**. Click **OK** to close the **Stream Lookup** edit properties dialog box.

Specify the fields to retrieve :

| #	Field	New name	Default	Type	
1	ZIP_RESOLVED			String	

9. Save your transformation.

In the canvas, you can select the **Lookup Missing Zips** step, right-click and select **Preview**. to display the preview/debugger dialog box. Click **Quick Launch** to preview the data flowing through this step. Notice that the new field, **ZIP_RESOLVED**, has been added to the stream containing your resolved postal codes.

Complete your Transformation

After you *resolve missing zip code information*, the last task is to clean up the field layout on your lookup stream. Cleaning up makes it so that it matches the format and layout of your other stream going to the **Write to Database** step. Create a **Select values** step for renaming fields on the stream, removing unnecessary fields, and more.

1. Add a **Select Values** step to your transformation by expanding the **Transform** folder and choosing **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.
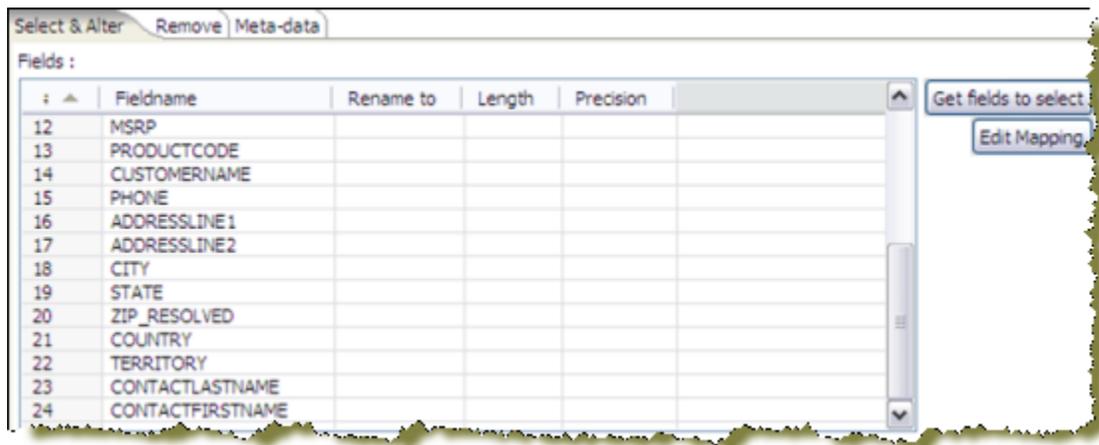
8. The original POSTALCODE field was formatted as an 9-character string. You must modify your new field to match the form. Click the **Meta-Data** tab.

9. In the first row of the **Fields to alter table**, click in the **Fieldname** column and select **ZIP_RESOLVED**.

10. Type **POSTALCODE** in the **Rename** to column; select **String** in the Type column, and type **9** in the **Length** column. Click **OK** to exit the edit properties dialog box.

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

12. Save your transformation as **Getting Started Transformation**.
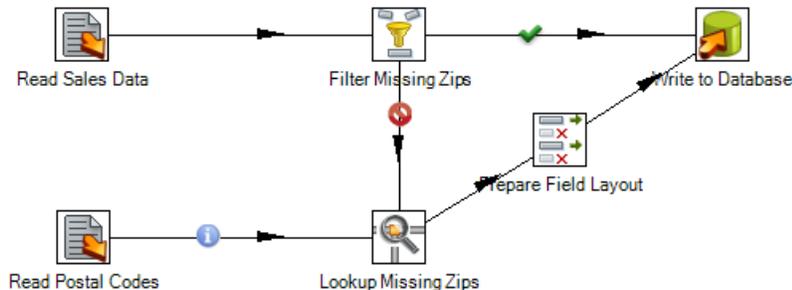
Run Your Transformation

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 your other needs. 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.

1. In the Spoon graphical interface, click **Run This Transformation**.
   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 canvas.

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.

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

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 works only 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, see *Create DI Solutions*.

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.

The **Metrics tab** allows you to see a Gantt charter after the transformation or job has run. This shows you information such as how long it takes to connect to a database, how much time is spent executing a SQL query, or how long it takes to load a transformation.
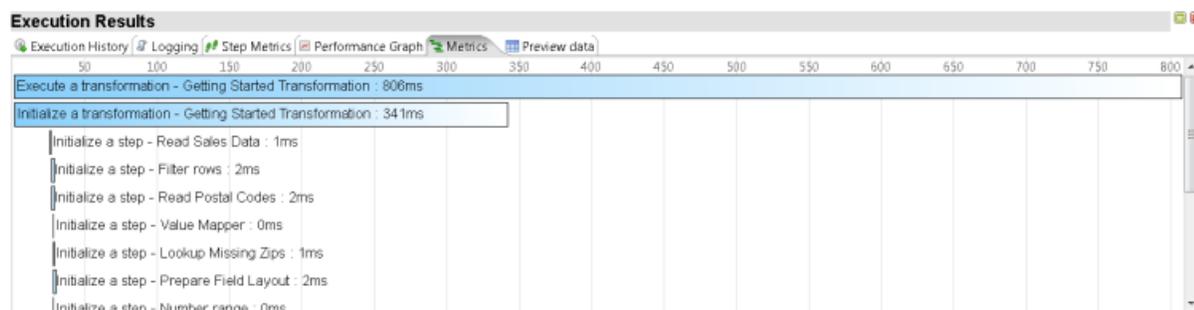
For more information on configuring logging or performance monitoring, see How to Enable Logging or the content relating to the Data Integration Operations Mart.

Create Jobs

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 and run your transformation to load the records into the database. In a subsequent exercise, you will schedule the job to be run every Sunday morning at 9 a.m.

To complete this exercise, you must have completed the exercises in the *Create Transformations* section.

1. Go to **File** > **New** > **Job**.
2. Expand the **General** folder and drag a **Start** job entry onto the graphical workspace.
   The **Start** job entry defines where the execution will begin.
3. Expand the **Conditions** folder and add a **File Exists** job entry.
4. Draw a hop from the **Start** job entry to the **File Exists** job entry.
5. 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: `...\design-tools\data-integration\samples \transformations\files`.

   Be sure to set the filter to CSV files to see the file.
6. Expand the **General** folder and add a **Transformation** job entry.
7. Draw a hop between the **File Exists** and the **Transformation** job entries.
8. Double-click the **Transformation** job entry to open its edit properties dialog box.
9. Select the **Specify by name and directory** option. Click **Browse**.
10. Expand the repository tree to find your sample transformation. Select it and click **OK**.
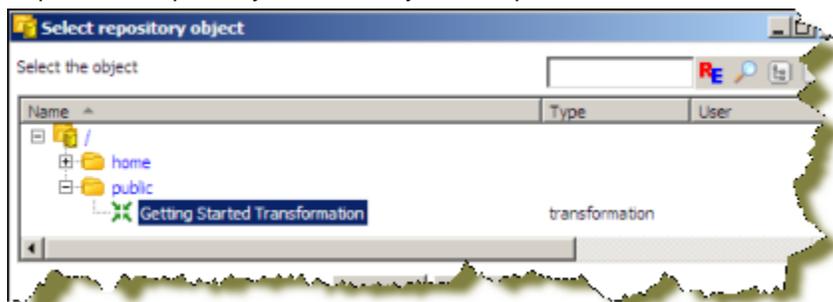
11. Create a hop between **File Exists** and **Getting Started Transformation**. Save your transformation as **Sample Job**.

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

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

Schedule Jobs

The Enterprise Edition of the DI Server provides scheduling services that allow 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 morning at 9 o'clock.

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

5. For the **End** date, select **Date** and then enter a date several weeks in the future using the calendar picker.
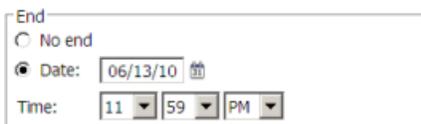
6. Click **OK** to complete your schedule.

   Note: The scheduler includes full support for PDI's parameters, arguments, and variables. For more detailed information on scheduling options, please refer to *Schedule and Script PDI Content*.

7. To view, edit and manage all scheduled activities, click *the Schedule perspective on the main toolbar*. In the **Schedule perspective** 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.
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.

   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.

Use 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 are the top 10 countries 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 the tab containing the sample transformation you just created in *Create Transformations*.
2. Right-click the **Write to Database** step, and select **Visualize** > **Analyzer**.
   In the background, PDI 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.

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

Immediately you can see that there is another problem with the quality of the data. 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 return to the **Data Integration** perspective to resolve this issue.

Correct Data Quality

Follow thees instructions 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** step to the **Value Mapper** step and select **Result is TRUE**.
5. Draw a hop from the **Prepare Field Layout** step to the **Value Mapper** step.
6. Draw a hop from the **Value Mapper** step to the **Write to Database** step.
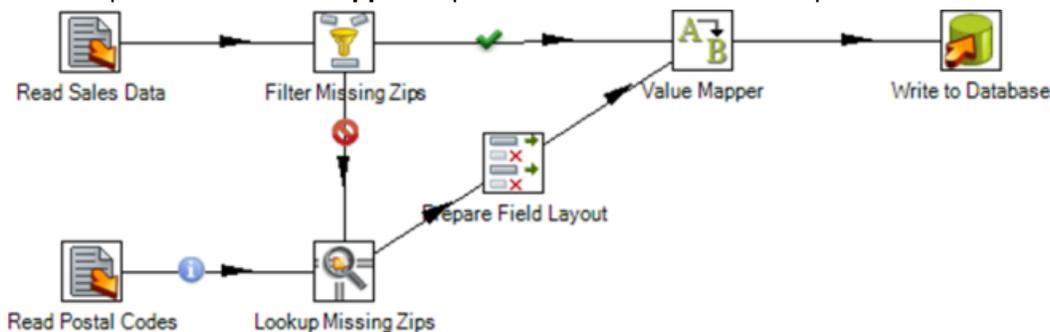
7. Double-click on the **Value Mapper** step to open its edit step properties dialog box.
8. Select the **COUNTRY** field in the **Fieldname** to use input.
9. In the first row of the **Field Values** table, type **United States** as the **Source** value and **USA** as the **Target value**. Click **OK** to exit the dialog box.
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.
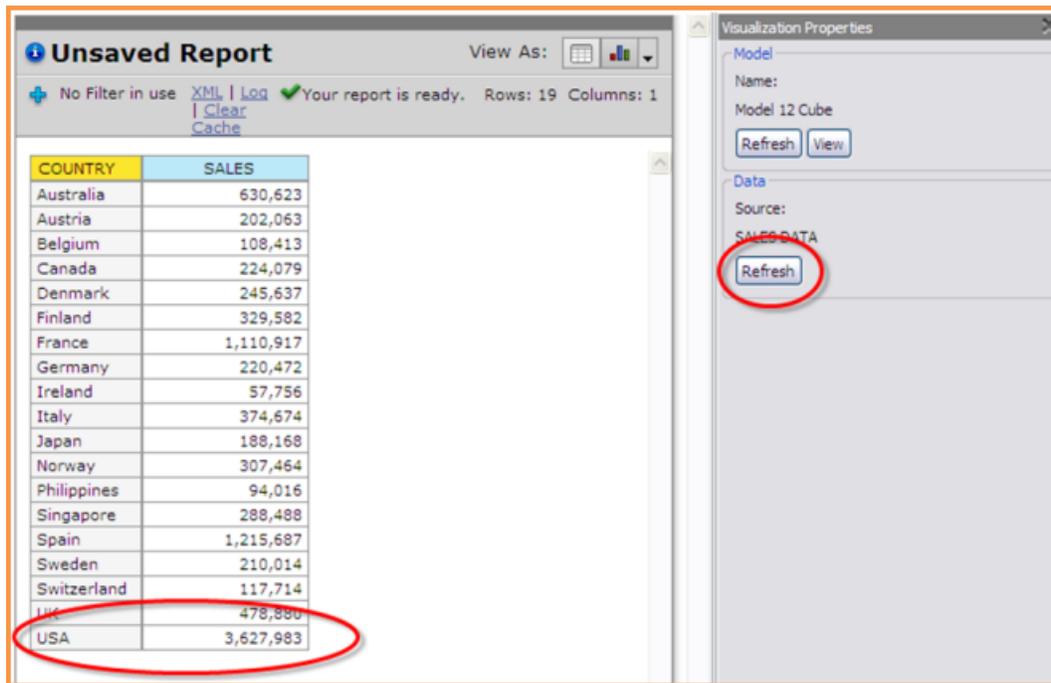
Create a Top Ten Countries by Sales Chart

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 **Stacked 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. Use the **Data Integration** perspective to add this 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 between the **Value Mapper** step and the **Write to Database** (Table Output) step. Click **Yes** to split the hop.
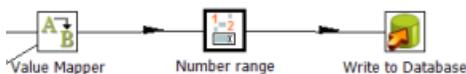
3. Double-click **Number range** to open its edit properties dialog box.
4. Choose the **SALES** field as your Input field.
5. Type **DEAL SIZE** as the name for the **Output** field.
6. In the **Ranges** table, define number ranges as shown in the example below. Click **OK**.

> **Note:** Because this step adds a new field into the stream, you must update your target database table to add the new column in the next steps.

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

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

9. 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.

10. Save and run your transformation to re-populate your target database.
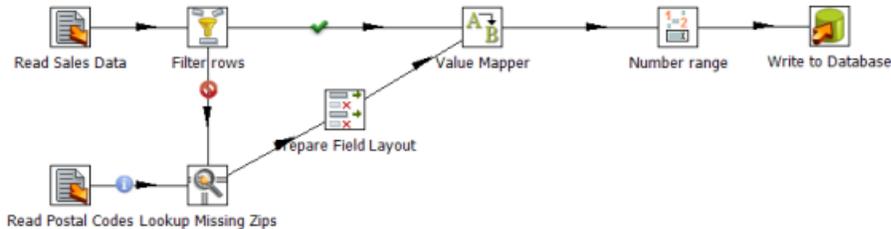
Wrapping it Up

Follow these instructions 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.
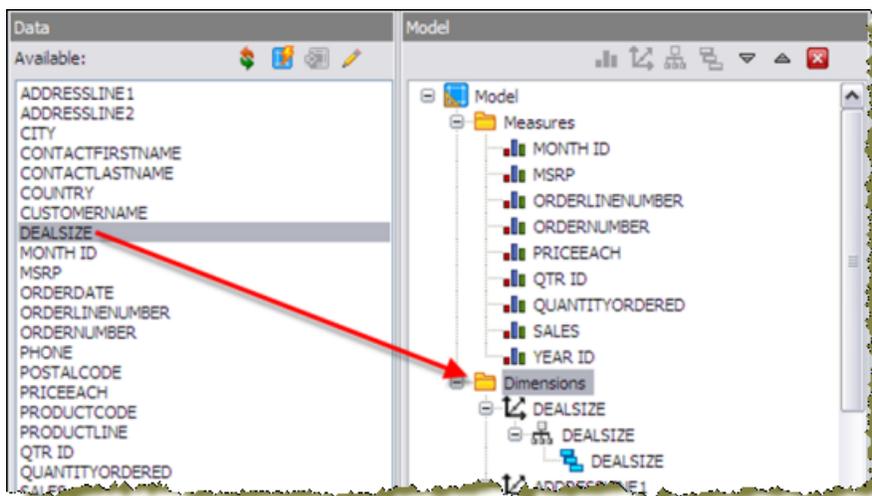
4. Click **Save** on the main toolbar to save your updated model. Click **Visualize** to return to your Top 10 Countries chart.

5. Click **Refresh** to update your field list to include the new **DEALSIZE** attribute.

6. Click **Toggle Layout** to open the **Layout** panel.

7. Drag **DEALSIZE** from the field list on the left into the **Color Stack** section of the **Layout** panel.

8. Click **Toggle Layout** to close the Layout Panel. You have successfully delivered your business user's request
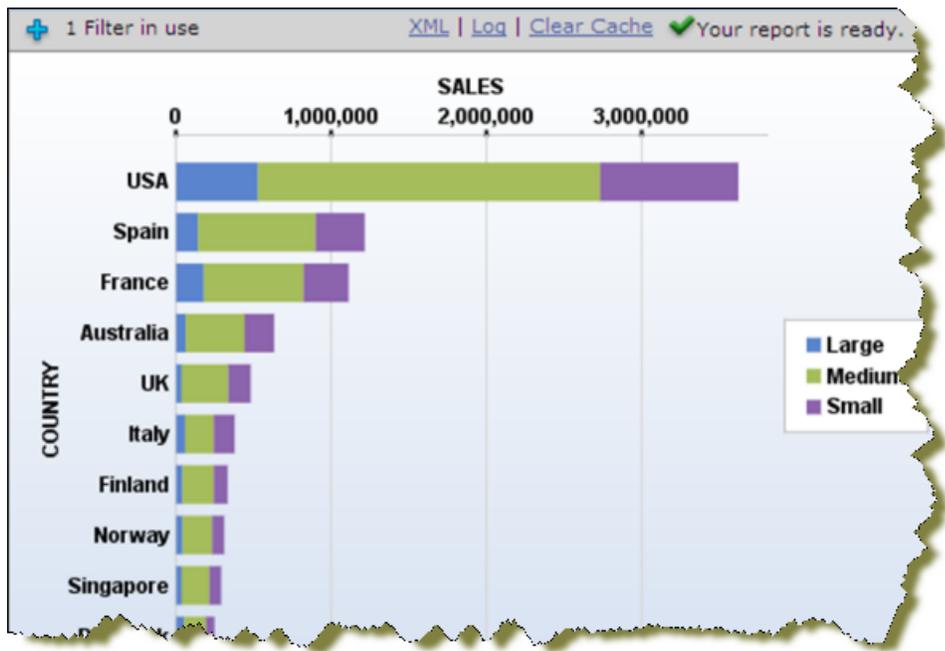
Getting Started with PDI and Hadoop

Pentaho provides a complete big data analytics solution that supports the entire big data analytics process. From big data aggregation, preparation, and integration, to interactive visualization, analysis, and prediction, Pentaho allows you to harvest the meaningful patterns buried in big data stores. Analyzing your big data sets gives you the ability to identify new revenue sources, develop loyal and profitable customer relationships, and run your organization more efficiently and cost effectively.

Pentaho, Big Data, and Hadoop

The term big data applies to very large, complex, or dynamic datasets that need to be stored and managed over a long time. To derive benefits from big data, you need the ability to access, process, and analyze data as it is being created. However, the size and structure of big data makes it very inefficient to maintain and process it using traditional relational databases.

Big data solutions re-engineer the components of traditional databases—data storage, retrieval, query, processing—and massively scales them.

Pentaho Big Data Overview

Pentaho increases speed-of-thought analysis against even the largest of big data stores by focusing on the features that deliver performance.

- **Instant access**—Pentaho provides visual tools to make it easy to define the sets of data that are important to you for interactive analysis. These data sets and associated analytics can be easily shared with others, and as new business questions arise, new views of data can be defined for interactive analysis.
- **High performance platform**—Pentaho is built on a modern, lightweight, high performance platform. This platform fully leverages 64-bit, multi-core processors and large memory spaces to efficiently leverage the power of contemporary hardware.
- **Extreme-scale, in-memory caching**—Pentaho is unique in leveraging external data grid technologies, such as Infinispan and Memcached to load vast amounts of data into memory so that it is instantly available for speed-of-thought analysis.
- **Federated data integration**—Data can be extracted from multiple sources, including big data and traditional data stores, integrated together and then flowed directly into reports, without needing an enterprise data warehouse or data mart.

About Hadoop

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

A Hadoop platform consists of a Hadoop kernel, a *MapReduce* model, a distributed file system, and often a number of related projects—such as *Apache Hive*, *Apache HBase*, and others.

A Hadoop Distributed File System, commonly referred to as HDFS, is a Java-based, distributed, scalable, and portable file system for the Hadoop framework.

Big Data Resources

- ***Pentaho Big Data Analytics Center***
- ***Pentaho Big Data Wiki***
- ***Apache Hadoop project*** -- A project that contains libraries that allows for the distributed processing of large data sets across clusters of computers using simple programming models. There are several modules, including the ***Hadoop Distributed File System (HDFS)***, which is a distributed file system that provides high-throughput access to application data and ***Hadoop MapReduce***, which is a key algorithm to distribute work around a cluster.

- *Avro*—A data serialization system
- *Cassandra*—A scalable multi-master database with no single points of failure
- *HBase*—A scalable, distributed database that supports structured data storage for large tables
- *Hive*—A data warehouse infrastructure that provides data summarization and on-demand querying
- *Pig*—A high-level, data-flow language and execution framework for parallel computation
- *ZooKeeper*—A high-performance coordination service for distributed applications
- *MongoDB*— A NoSQL open source document-oriented database system developed and supported by 10gen
- *Splunk* - A data collection, visualization and indexing engine for operational intelligence that is developed by Splunk, Inc.
- *CouchDB*—A NoSQL open source document-oriented database system developed and supported by Apache
- *Sqoop*—Software for transferring data between relational databases and Hadoop
- *Oozie*—A workflow scheduler system to manage Hadoop jobs

Why Choose Enterprise Edition?

Enterprise Edition enables you to deploy Pentaho Data Integration with confidence, security, and far lower total cost of ownership than proprietary and open source alternatives. Benefits of Pentaho Data Integration Enterprise Edition include:

Professional, Technical Support

- Live support provided by a knowledgeable team of product experts that consistently rates higher in customer satisfaction than the BI megavendors
- Dedicated customer case triage providing faster response times and increased priority for customer reported defects and enhancements

Enterprise Edition Features

- Enterprise security with granular control over content and actions that can be performed by users and roles. Enterprise security can be managed directly in Pentaho Data Integration Enterprise Edition or configured to integrate with your existing LDAP or Active Directory implementation
- Centralized content management facilitating team collaboration including secured sharing of content, content versioning (revision history), and transformation and job locking
- Integrated scheduler allowing you to schedule job and transformations for future or recurring execution; schedules are created and managed directly in the easy-to-use, graphical designer (Spoon)
- Additional transformation steps and job entries for integrating with third-party applications, messaging architectures and more

Certified Software Releases

- All certified software releases go through rigorous quality testing and a managed release process to ensure the stability of your production deployments
- Only subscription customers get access to maintenance releases containing critical defect resolutions

  Note: Binary distributions of Community Editions are provided with major product releases only. If you have a critical defect or improvement that has been addressed as part of a minor or patch release, you must wait for and upgrade to the next major release of Pentaho Data Integration.

Pricing for Pentaho Data Integration Enterprise Edition can be found at *http://www.pentaho.com/explore/how-to-buy/*. For more information or to start your subscription today, contact us at *http://www.pentaho.com/contact/*.