Getting Started with Pentaho Data Integration

Help and Support Resources

If you have questions that are not covered in this guide, or if you would like to report errors in the documentation, 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 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

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.

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

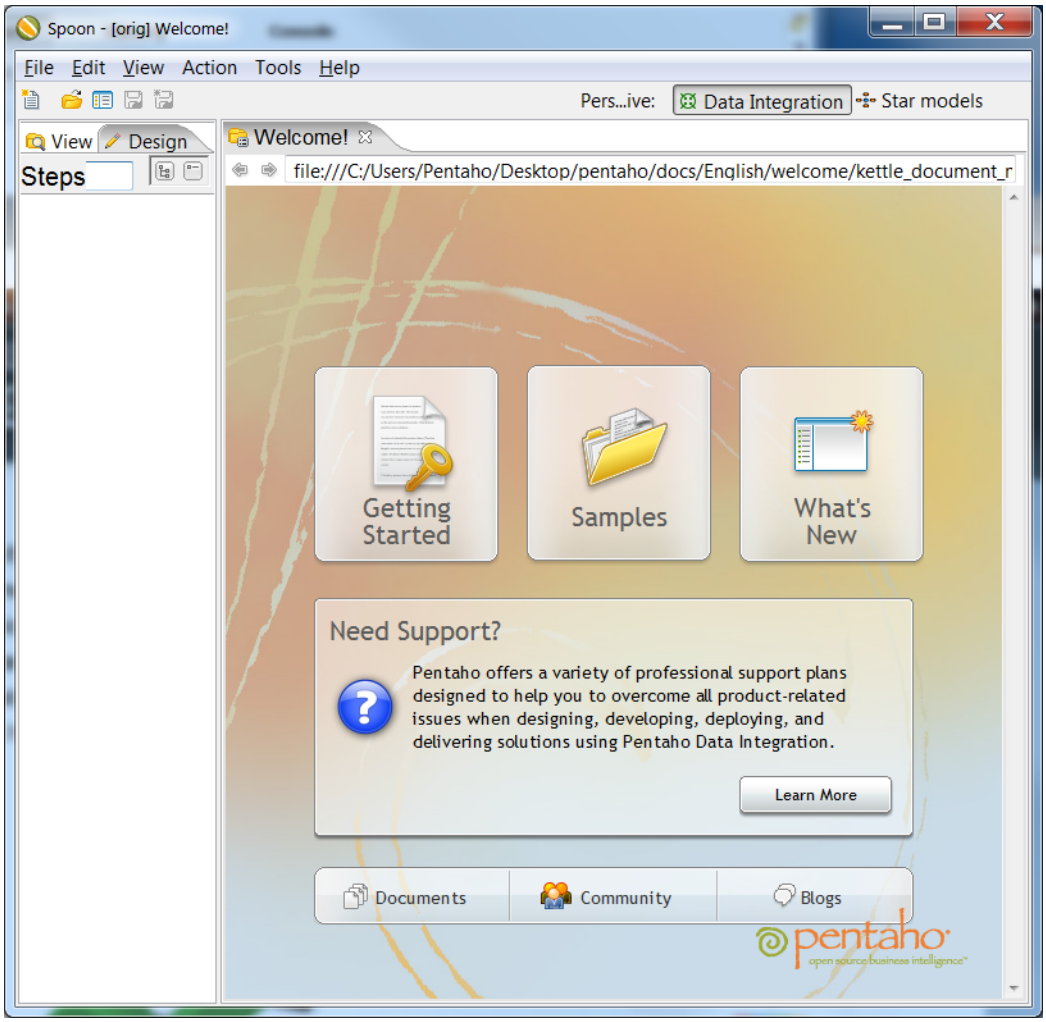
Documentation Suggestions: documentation@pentaho.com

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

Contents

Introduction

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



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

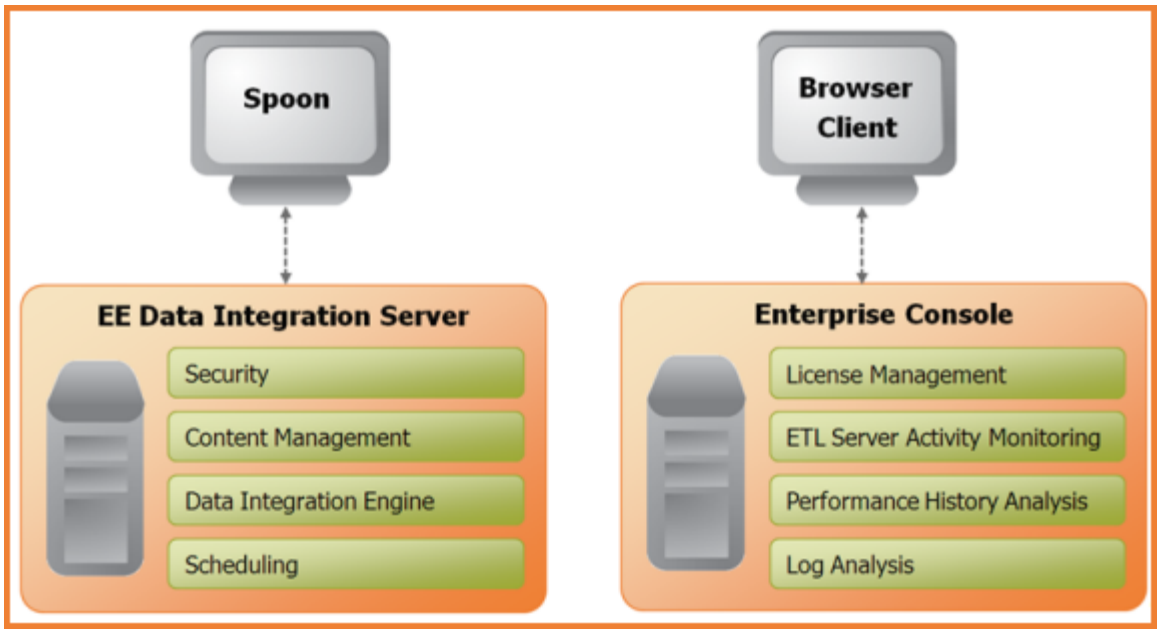
Pentaho Data Integration features and benefits include:

- 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

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 activities 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 *Java 6.0* already installed.

1. Go to the *Pentaho Product Download page*.
2. Select the appropriate operating system requirements.

Note: The installation instructions in this document are based on the Windows Operating System exclusively.

3. Fill out the contact form and hit send. Your download should start automatically.

Installing Pentaho Data Integration

It is assumed that you will follow the default installation instructions and that you are installing to a local device (localhost).

1. Read and accept the License Agreement.
2. Specify the location where you want to install Pentaho Data Integration or click **Next** to accept the default.
3. Set the user name and password for the Administrator account. For the purposes of this evaluation, accept the default user name, "admin," and type "password" in **Password** and **Confirm Password** fields.
4. Click **Next** to accept the default installation options on the **Summary** page.
5. Click **Next** to begin installation.

Pentaho Data Integration is installed as a Window service. When installation is complete, the Spoon designer is launched.

Starting the Spoon Client Tool

If you inadvertently exit Spoon, follow the instructions below to launch it again.

1. Navigate to the folder where you have installed Pentaho Data Integration; for example **c:\Program Files\pentaho \design-tools\data-integration**.
2. Double-click **Spoon.bat** to launch the designer.

   Note: If you are using Linux, double-click **spoon.sh**. To start the Spoon Designer on a Mac, go to `.../pdi-ee/data-integration` and double click on the **Data Integration 32-bit** or **Data Integration 64-bit** icon depending on your system.

3. Alternatively, in Windows, go to **Start** -> **Pentaho Enterprise Edition** -> **Design Tools** to launch the designer.

Starting the Data Integration Server

Follow the directions below to start the DI Server.

1. Navigate to the Pentaho Data Integration installation directory.

   ```
   cd c:\Program Files\pentaho\server\data-integration-server\
   ```

2. Run the **start-pentaho.bat** script to start the DI Server.
3. Alternatively, you can use the Start menu. Go to the **Server Control** section of the **Pentaho Enterprise Edition** Start menu folder, then click on **Start Data Integration Server**.

Pentaho Data Integration Folders and Scripts

After installation, your **pentaho** folder contains the following files and directories:

File/Folder Name	Description
\design-tools\data-integration	Contains the Spoon designer and command line utilities
\server\data-integration-server	Contains the data integration server including individual start/stop scripts; contains the enterprise console server including individual start/stop scripts
\design-tools\docs\English	Contains this document
\server\data-integration-server\ start-pentaho.bat	Script file for starting the Data Integration server on Windows

File/Folder Name	Description
\server\data-integration-server\ start-pentaho.sh	Script file for starting the Data Integration server on Linux and Macintosh
\server\data-integration-server\ stop-pentaho.bat	Script file for stopping the Data Integration server on Windows
\server\data-integration-server\ stop-pentaho.sh	Script file for stopping the Data Integration server on Linux and Macintosh
\design-tools\data-integration\Spoon.bat	Script file for starting the Spoon Designer on Windows
\design-tools\data-integration\spoon.sh	Script file for starting the Spoon Designer on Linux and Macintosh

Adding a JDBC Driver

Before you can connect to a data source in any Pentaho server or client tool, you must first install the appropriate database driver. Your database administrator, CIO, or IT manager should be able to provide you with the proper driver JAR. If not, you can download a JDBC driver JAR file from your database vendor or driver developer's Web site. Once you have the JAR, follow the instructions below to copy it to the driver directories for all of the Business Analytics components that need to connect to this data source.

Note: Microsoft SQL Server users frequently use an alternative, non-vendor-supported driver called JTDS. If you are adding an MSSQL data source, ensure that you are installing the correct driver.

Backing up old drivers

You must also ensure that there are no other versions of the same vendor's JDBC driver installed in these directories. If there are, you may have to back them up and remove them to avoid confusion and potential class loading problems. This is of particular concern when you are installing a driver JAR for a data source that is the same database type as your Pentaho solution repository. If you have any doubts as to how to proceed, contact your Pentaho support representative for guidance.

Installing JDBC drivers

Copy the driver JAR file to the following directories, depending on which servers and client tools you are using (Dashboard Designer, ad hoc reporting, and Analyzer are all part of the BA Server):

Note: For the DI Server: before copying a new JDBC driver, ensure that there is not a different version of the same JAR in the destination directory. If there is, you must remove the old JAR to avoid version conflicts.

- **BA Server:** `/pentaho/server/biserver-ee/tomcat/lib/`
- **Enterprise Console:** `/pentaho/server/enterprise-console/jdbc/`
- **Data Integration Server:** `/pentaho/server/data-integration-server/tomcat/webapps/pentaho-di/WEB-INF/lib/`
- **Data Integration client:** `/pentaho/design-tools/data-integration/libext/JDBC/`
- **Report Designer:** `/pentaho/design-tools/report-designer/lib/jdbc/`
- **Schema Workbench:** `/pentaho/design-tools/schema-workbench/drivers/`
- **Aggregation Designer:** `/pentaho/design-tools/agg-designer/drivers/`
- **Metadata Editor:** `/pentaho/design-tools/metadata-editor/libext/JDBC/`

Note: To establish a data source in the Pentaho Enterprise Console, you must install the driver in both the Enterprise Console and the BA Server or Data Integration Server. If you are just adding a data source through the Pentaho User Console, you do not need to install the driver to Enterprise Console.

Restarting

Once the driver JAR is in place, you must restart the server or client tool that you added it to.

Connecting to a Microsoft SQL Server using Integrated or Windows Authentication

The JDBC driver supports Type 2 integrated authentication on Windows operating systems through the **integratedSecurity** connection string property. To use integrated authentication, copy the **sqljdbc_auth.dll** file to all the directories to which you copied the JDBC files.

The **sqljdbc_auth.dll** files are installed in the following location:

```
<installation directory>\sqljdbc_<version>\<language>\auth\
```

Note: Use the **sqljdbc_auth.dll** file, in the x86 folder, if you are running a 32-bit Java Virtual Machine (JVM) even if the operating system is version x64. Use the **sqljdbc_auth.dll** file in the x64 folder, if you are running a 64-bit JVM on a x64 processor. Use the **sqljdbc_auth.dll** file in the IA64 folder, you are running a 64-bit JVM on an Itanium processor.

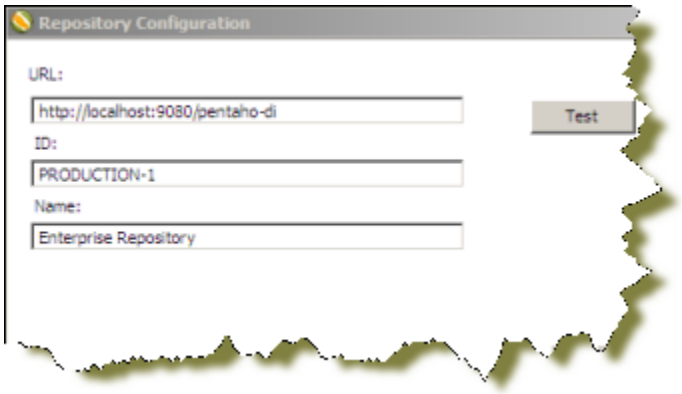
Connecting to the Enterprise Repository

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. Keep the default URL.

   The URL used to connect to the Data Integration server is provided by default.
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. Enter an **ID** and **Name** for your repository.



7. Click **OK** to exit the Repository Configuration dialog box.
   Your new connection appears in the list of available repositories.
8. Log on to the Enterprise Repository by entering the following credentials: user name = **joe**, password = **password**.

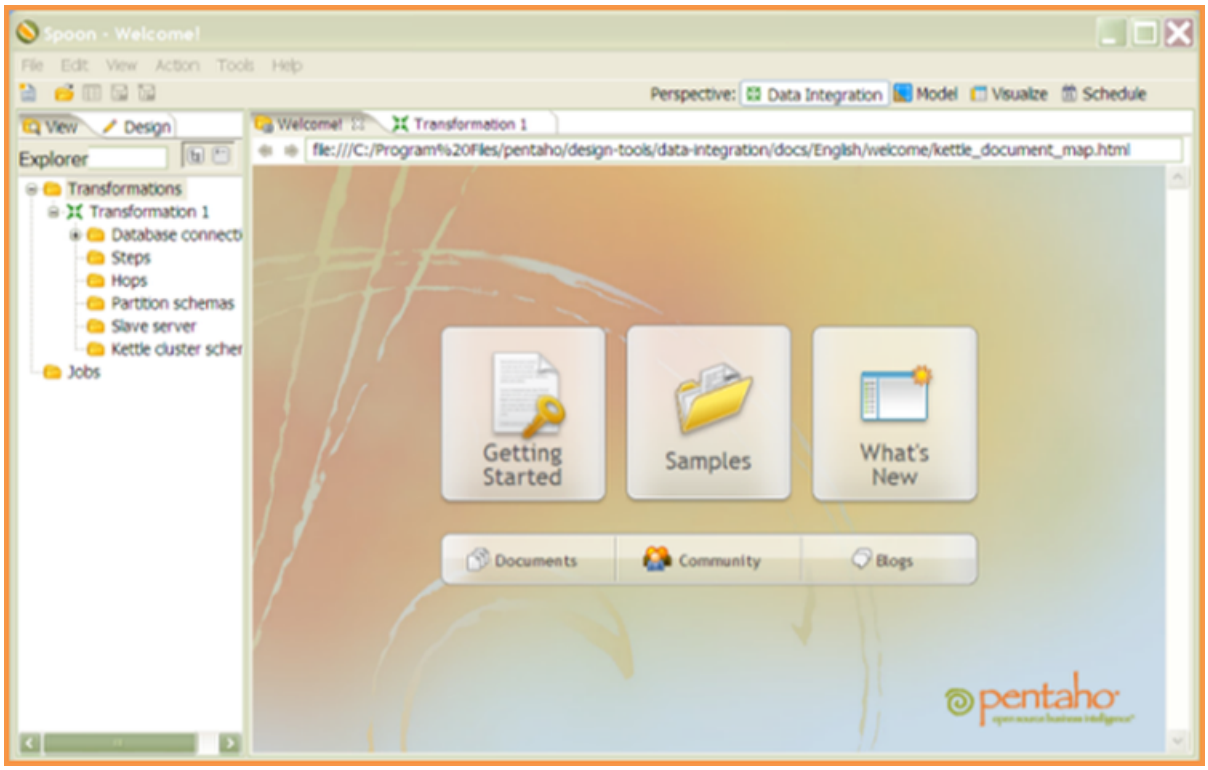
   The Data Integration Server is configured out of the box to use the Pentaho default security provider. This has been pre-populated with a set of sample users and roles including:

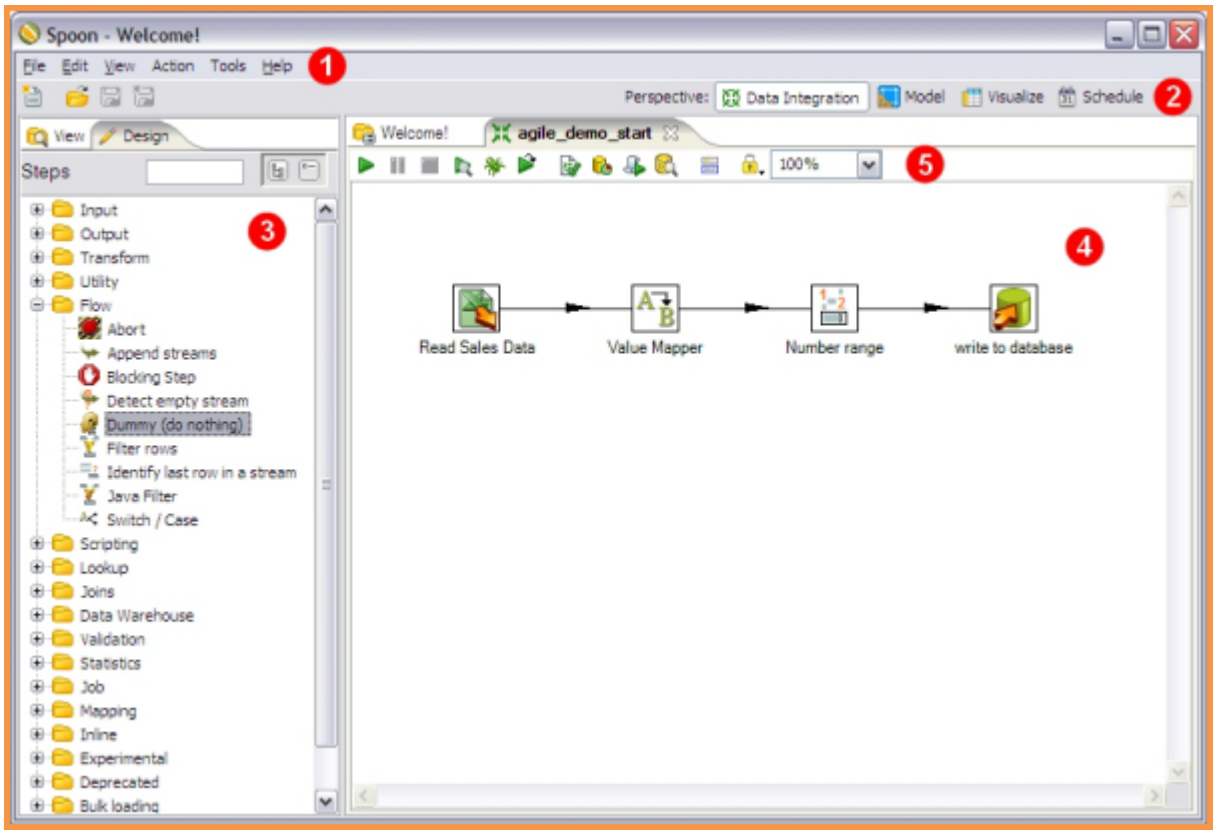
   - Joe — Member of the admin role with full access and control of content on the Data Integration Server
   - Suzy — Member of the CEO role with permission to read and create content, but not administer security

Note: See the *Pentaho Business Analytics Security Guide* available in the Pentaho InfoCenter 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.





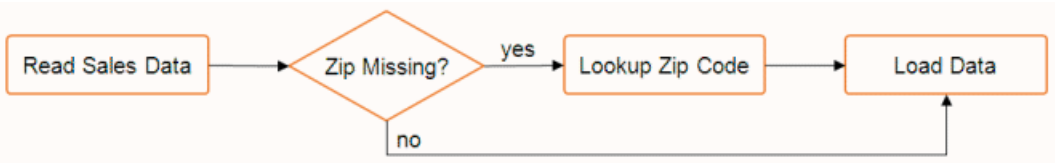
The Spoon Designer is organized into the components described in the table below:

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 BA 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?"

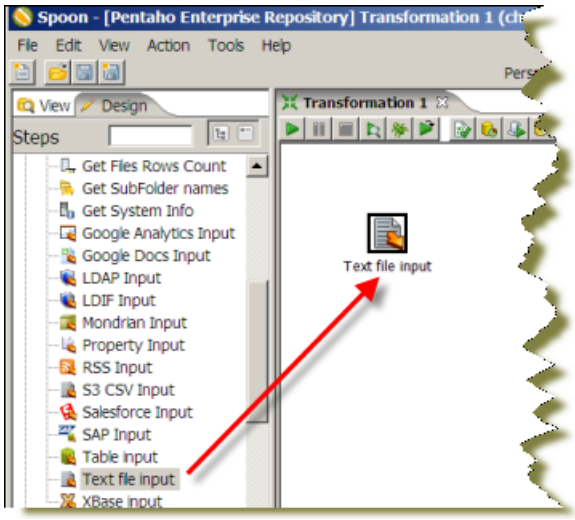
This exercise will step you through building your first transformation with Pentaho Data Integration introducing common concepts along the way. The exercise scenario includes a flat file (CSV) of sales data that you will load into a database so that mailing lists can be generated. Several of the customer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



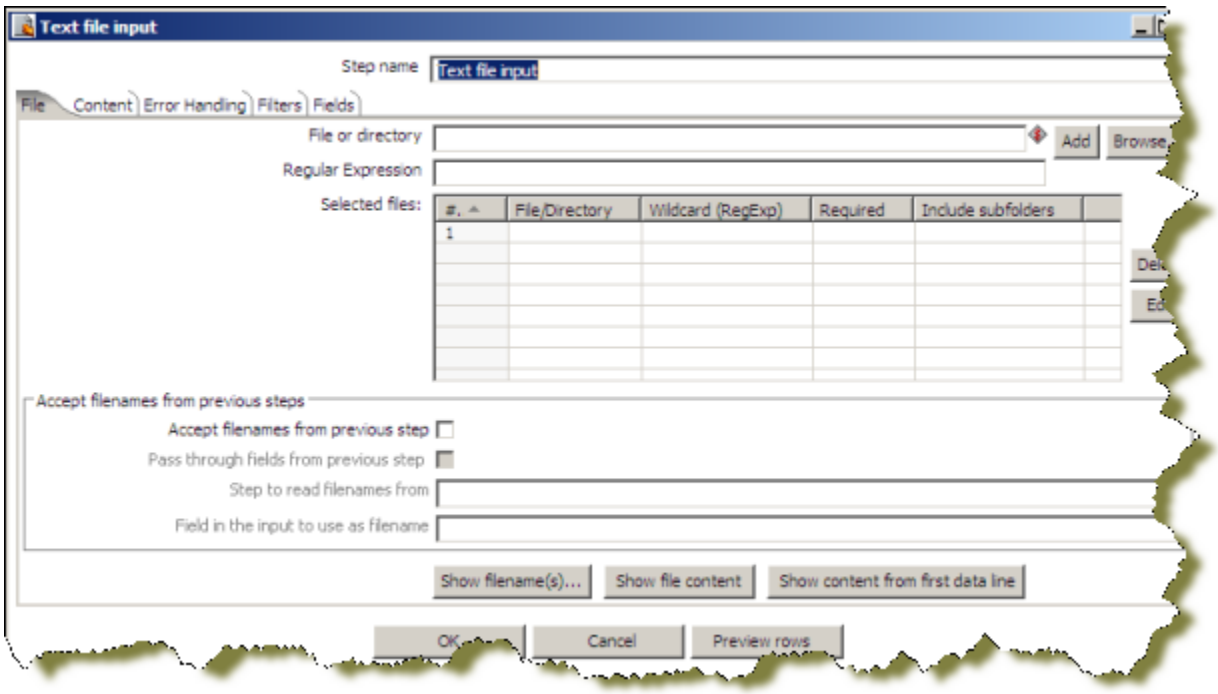
Retrieving Data from a Flat File (Text File Input Step)

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

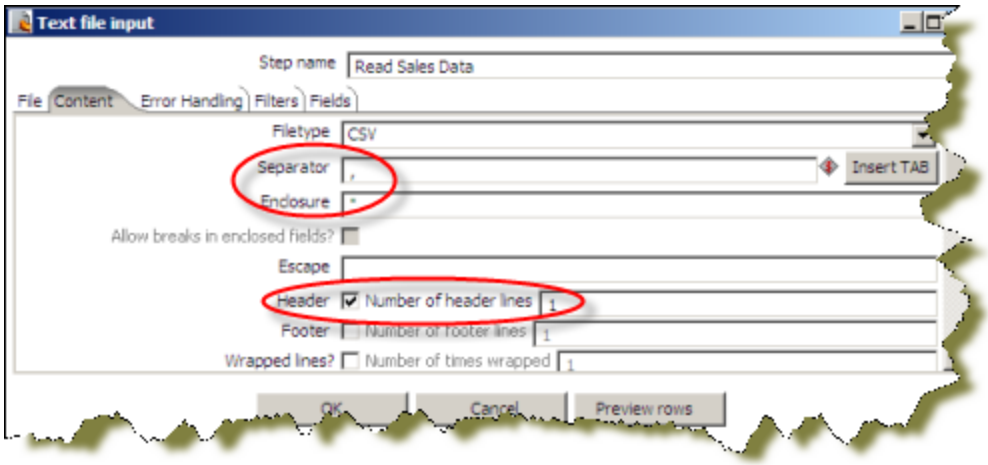
1. Click (New) in the upper left corner of the Spoon graphical interface.
2. Select **Transformation** from the list.
3. Under the **Design** tab, expand the **Input** node; then, select and drag a **Text File Input** step onto the canvas on the right.



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.

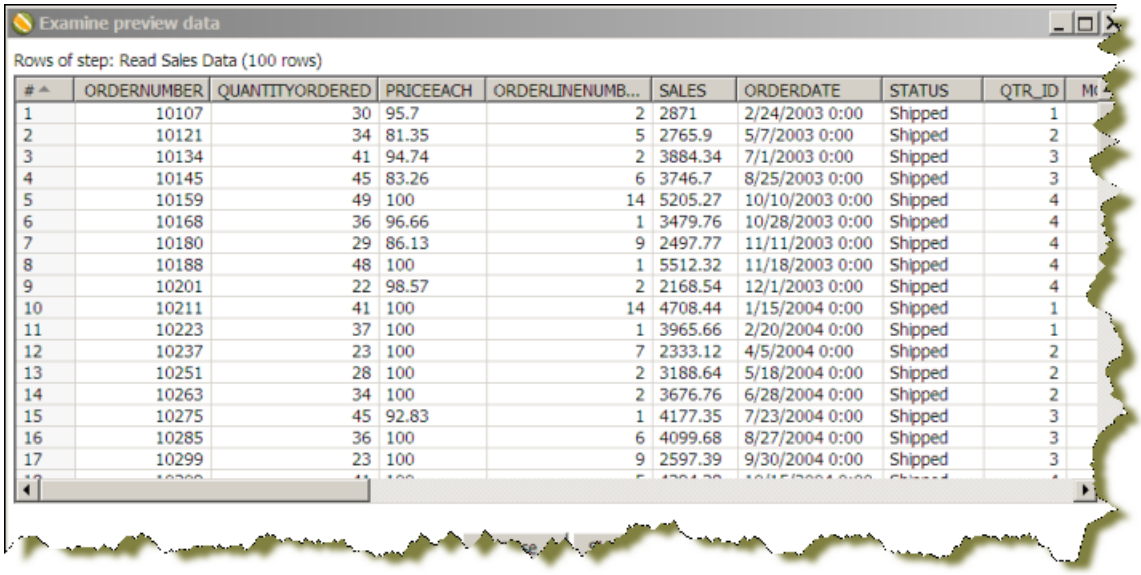


5. In the **Step Name** field, type **Read Sales Data**.
   You are renaming the Text File Input step to Read Sales Data.
6. Click **Browse** to locate the source file, **sales_data.csv**, available at `...\design-tools\data-integration\samples\transformations\files`.
   The path to the source file appears 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 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.



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 to scan, allowing you to determine default settings for the fields such as their format, length, and precision. Type **0** (zero) in the **Number of Sample Lines** text box to scan all lines. By scanning all lines, you ensure that Pentaho Data Integration has read the entire contents of the file and you reduce the possibility of errors that may cause a transformation not to run. Click **OK** and the summary of the scan results appears. Once you are done examining the scan results,click **Close** to return to the step properties editor.

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.



#	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMB...	SALES	ORDERDATE	STATUS	QTR_ID	M
1	10107	30	95.7	2	2871	2/24/2003 0:00	Shipped	1	
2	10121	34	81.35	5	2765.9	5/7/2003 0:00	Shipped	2	
3	10134	41	94.74	2	3884.34	7/1/2003 0:00	Shipped	3	
4	10145	45	83.26	6	3746.7	8/25/2003 0:00	Shipped	3	
5	10159	49	100	14	5205.27	10/10/2003 0:00	Shipped	4	
6	10168	36	96.66	1	3479.76	10/28/2003 0:00	Shipped	4	
7	10180	29	86.13	9	2497.77	11/11/2003 0:00	Shipped	4	
8	10188	48	100	1	5512.32	11/18/2003 0:00	Shipped	4	
9	10201	22	98.57	2	2168.54	12/1/2003 0:00	Shipped	4	
10	10211	41	100	14	4708.44	1/15/2004 0:00	Shipped	1	
11	10223	37	100	1	3965.66	2/20/2004 0:00	Shipped	1	
12	10237	23	100	7	2333.12	4/5/2004 0:00	Shipped	2	
13	10251	28	100	2	3188.64	5/18/2004 0:00	Shipped	2	
14	10263	34	100	2	3676.76	6/28/2004 0:00	Shipped	2	
15	10275	45	92.83	1	4177.35	7/23/2004 0:00	Shipped	3	
16	10285	36	100	6	4099.68	8/27/2004 0:00	Shipped	3	
17	10299	23	100	9	2597.39	9/30/2004 0:00	Shipped	3	

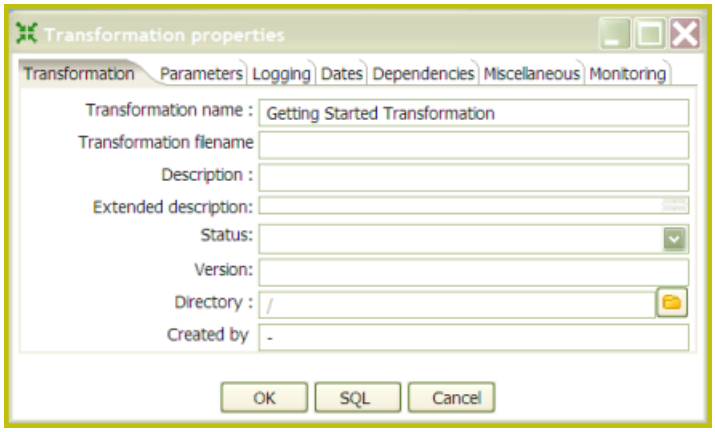
12. Save your transformation. See *Saving Your Transformation* on page 16.

Saving Your Transformation

Follow the instructions below to save your transformation.

Note: You can save your transformation at any point in this walk through. Saving allows you to start and stop the exercises at your convenience.

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**.

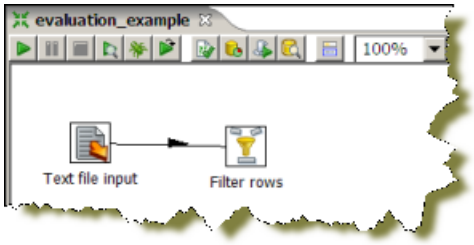


3. In the **Directory** field, click (folder icon) to select a repository folder where you will save your transformation.
4. Expand the **Home** directory and double-click the **joe** folder.
   Your transformation will be stored in the joe folder in the Enterprise Repository.
5. Click **OK** to exit 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 Enterprise Repository.
7. Click **OK** to exit the **Enter Comment** dialog box.

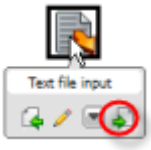
Filter Records with Missing Postal Codes (Filter Rows Step)

The source file contains several records that are 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.

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.

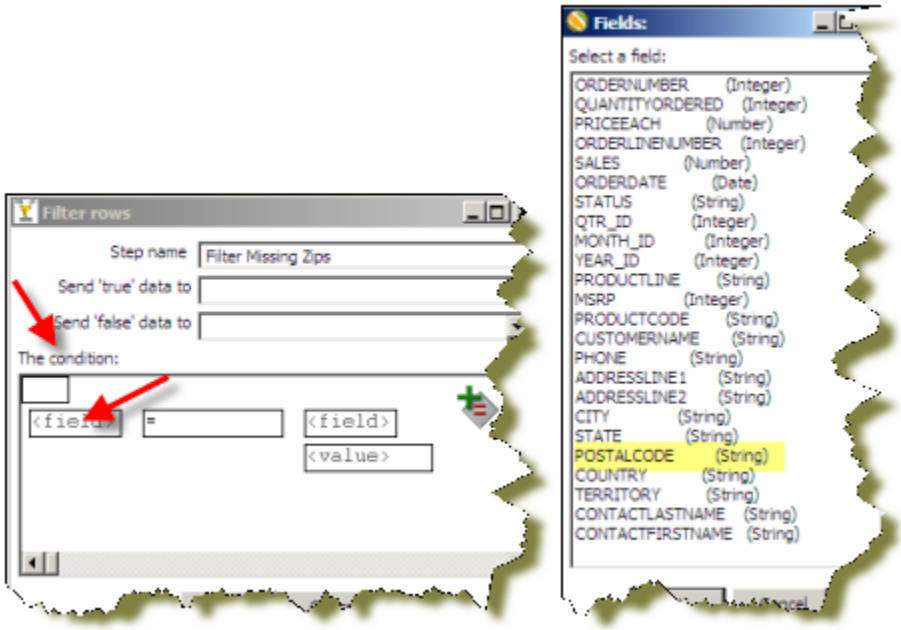


Alternatively, you can draw hops by hovering over a step until the hover menu appears. Drag the hop painter icon from the source step to your target step.

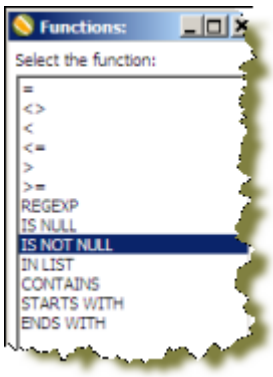


Note: For more information on hops including a description of color coding and hop icons, see the *Pentaho Data Integration User Guide* in the Pentaho InfoCenter.

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>`. A dialog box that contains the fields you can use to create your condition appears.



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.

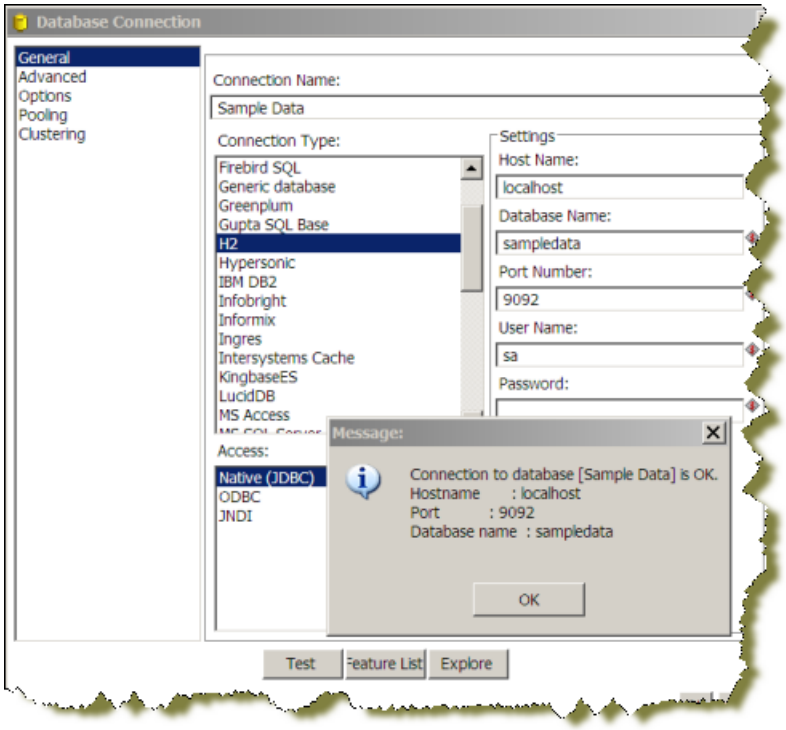
8. Save your transformation.

Loading Your Data into a Relational Database (Table Output Step)

In this exercise you will 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. 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 as shown in the table below.

Connection Name	Type, **Sample Data**
Connection Type:	Choose, H2
Host Name	localhost
Database Name	Type **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 Pentaho Data Integration* 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 **Write to Database** edit properties dialog box, enable the **Truncate Table** property.
11. Click **SQL** to generate the DDL for creating your 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 **OK** to close the Table Output edit properties dialog box.
13. Save your transformation.

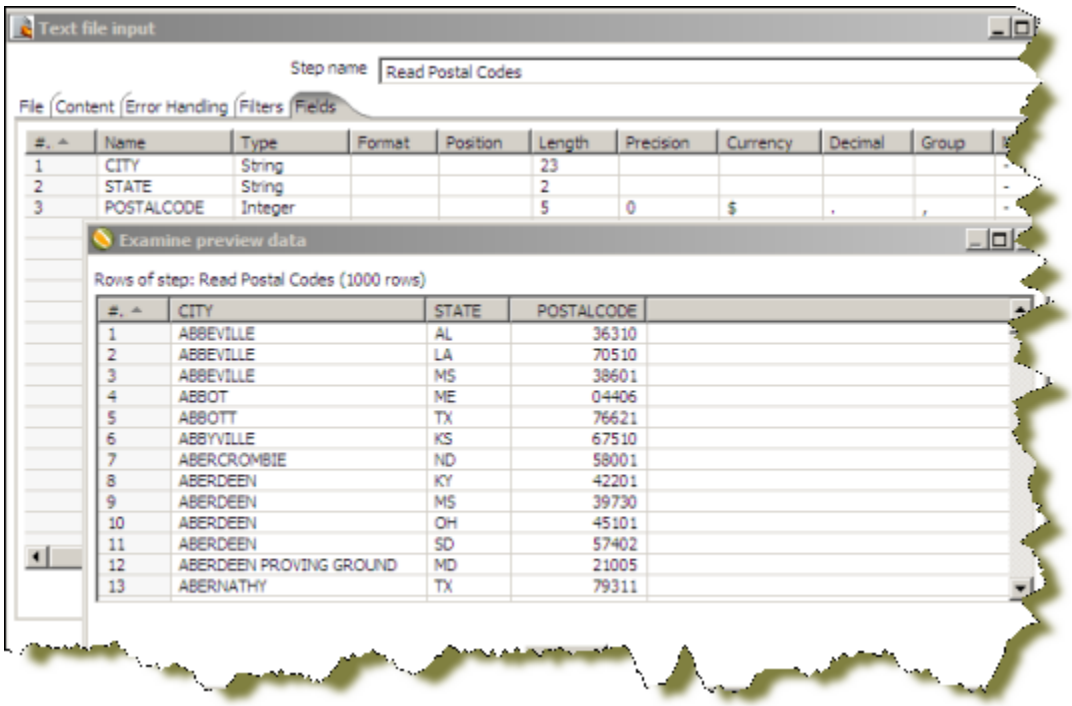
Retrieving Data from your Lookup File (Text File Input Step)

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, next 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, 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.

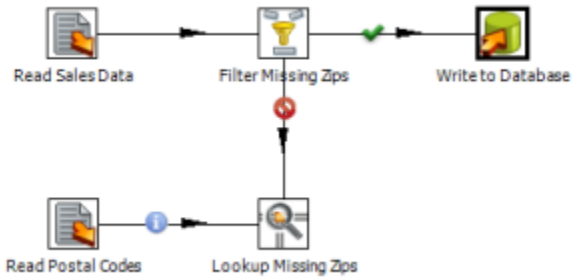


8. Save your transformation.

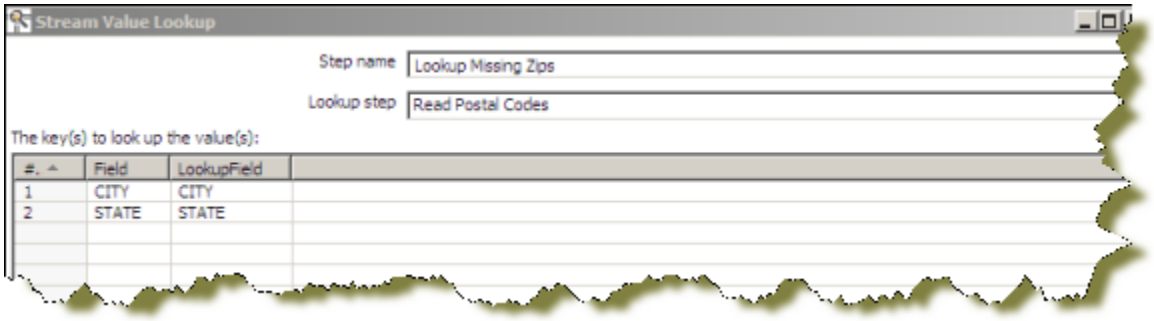
Resolving Missing Zip Code Information (Stream Lookup Step)

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

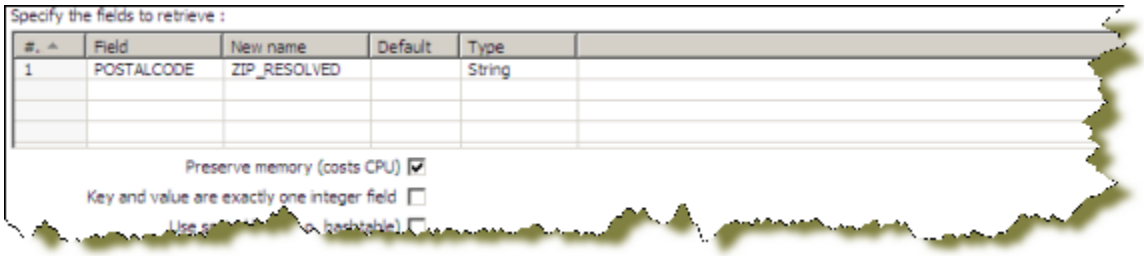
1. Add a **Stream Lookup** step to your transformation. Under the Design tab, expand the **Lookup** folder and choose **Stream Lookup**.
2. Draw a hop between the **Filter Missing Zips** (Filter rows) and **Stream Lookup** steps. Select the **Result is FALSE**.
3. Create a hop from the **Read Postal Codes** step (Text File input) to the Stream lookup step.
4. Double-click on the **Stream lookup** step to open its edit properties dialog box.
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 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:



8. Click **Get Lookup Fields**. **POSTALCODE** is the only field you want to retrieve. (To delete the extra 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.



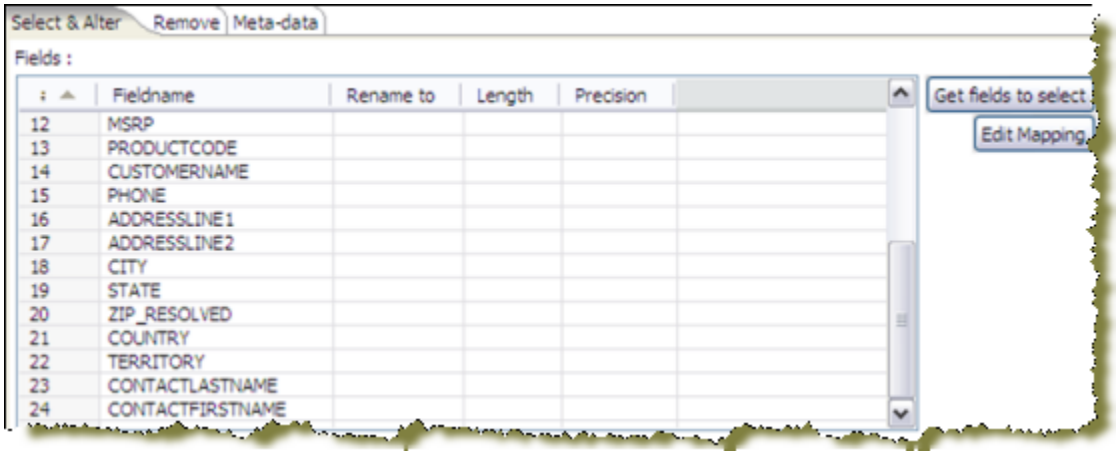
9. Save your transformation.

   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 that the new field, ZIP_RESOLVED, has been added to the stream containing your resolved postal codes.

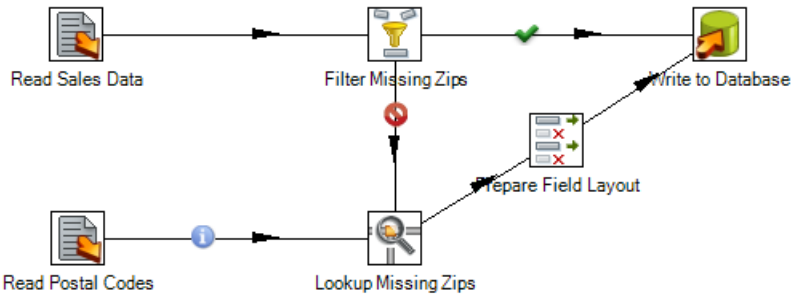
Completing your Transformation (Select Values Step)

The last task is to clean up the field layout on your lookup stream so that it matches the format and layout of your 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.



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.



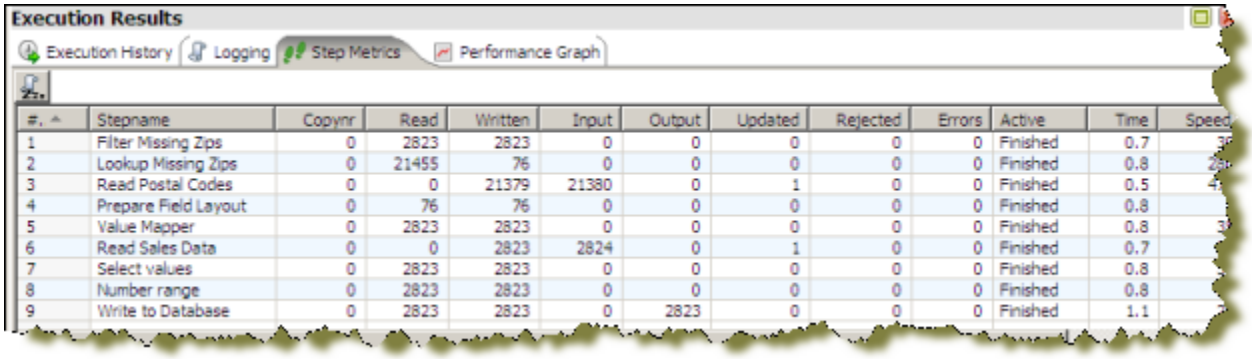
Running Your Transformation

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.

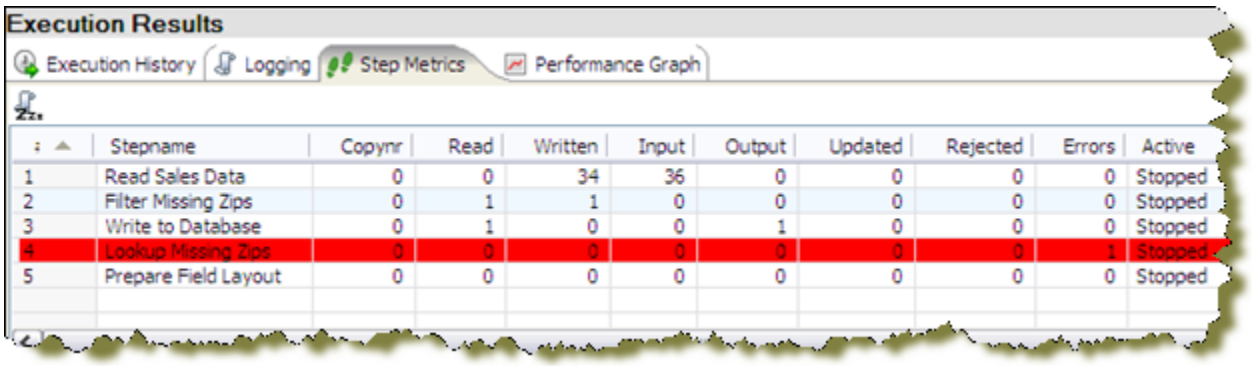
This final part of the creating a transformation exercise focuses exclusively on the local execution option. For more information on remote, clustered and other execution options review the links in the additional resources section later in this guide or in the *Pentaho Data Integration User Guide* in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

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.



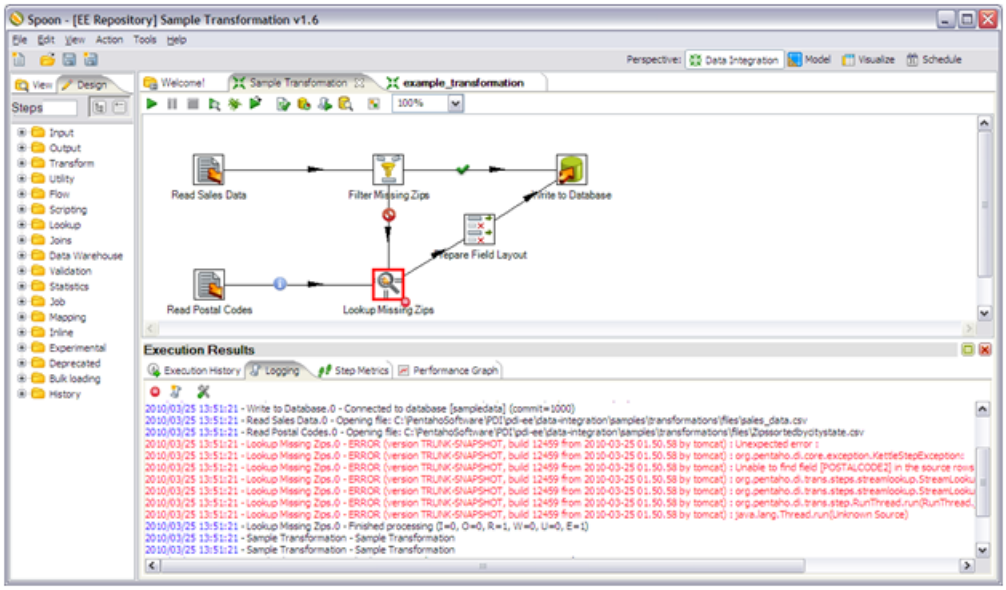
#.	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed
1	Filter Missing Zips	0	2823	2823	0	0	0	0	0	Finished	0.7	3
2	Lookup Missing Zips	0	21455	76	0	0	0	0	0	Finished	0.8	2
3	Read Postal Codes	0	0	21379	21380	0	1	0	0	Finished	0.5	4
4	Prepare Field Layout	0	76	76	0	0	0	0	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.



	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

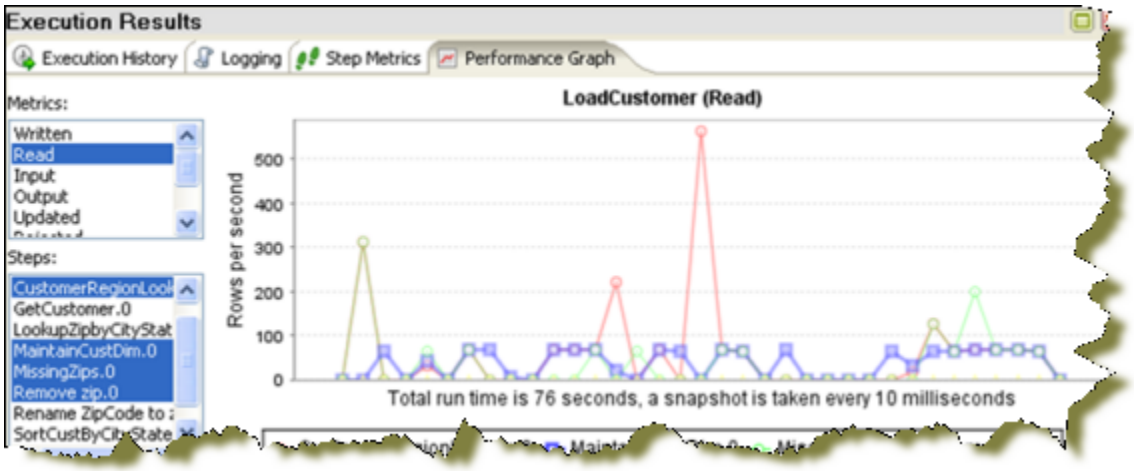
The **Logging** tab displays the logging details for the most recent execution of the transformation. Error lines are highlighted in red..



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 the *Pentaho Data Integration User Guide* found in the Pentaho InfoCenter.

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, see the *Pentaho Data Integration User Guide* found in the Pentaho InfoCenter.

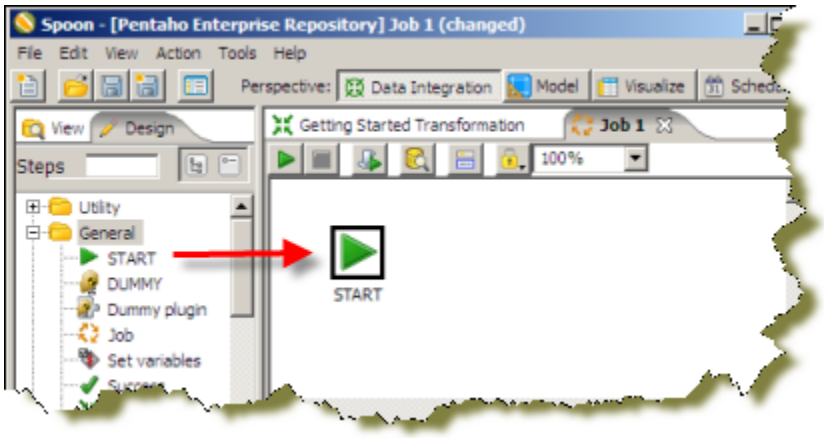
Building Your First Job

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

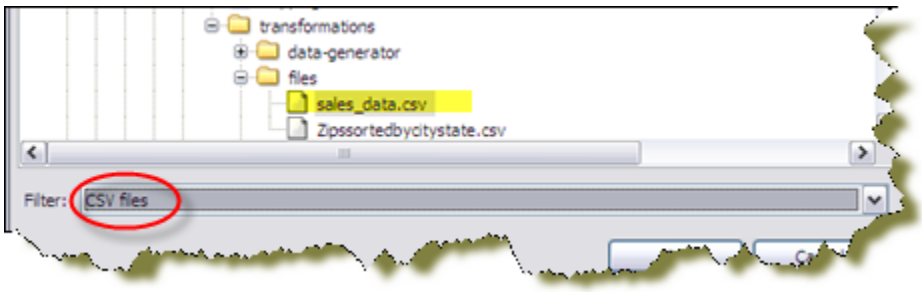
For this exercise, imagine that an external system is responsible for placing your **sales_data.csv** input in its source location every Saturday night at 9 p.m. You want to create a job that will check to see that the file has arrived and run your transformation to load the records into the database.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

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..

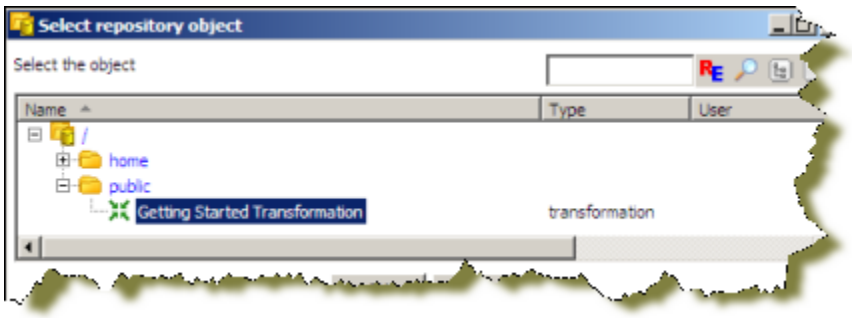


   The Start job entry defines where the execution will begin.
4. Expand the **Conditions** folder and add a **File Exists** job entry.
5. Draw a hop from the **Start** job entry to the **File Exists** job entry.
6. Double-click the **File Exists** job entry to open its edit properties dialog box. Click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

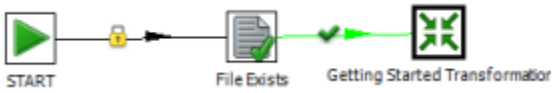
   Be sure to set the filter to CSV files to see the file.



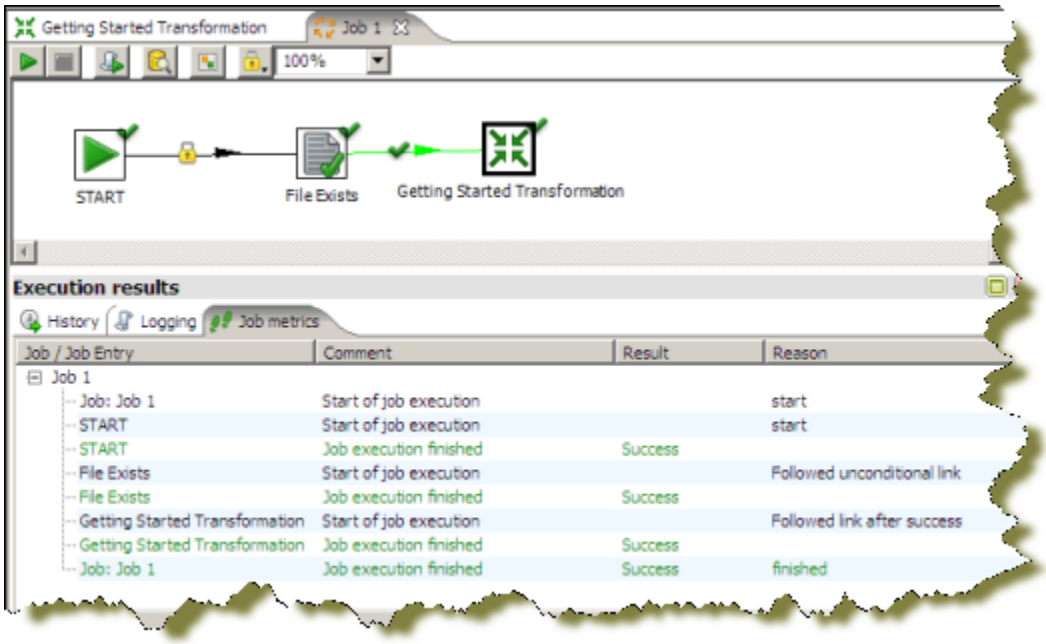
7. Expand the **General** folder and add a **Transformation** job entry.
8. Draw a hop between the **File Exists** and the **Transformation** job entries.
9. Double-click the **Transformation** job entry to open its edit properties dialog box.
10. Select the **Specify by name and directory** option. 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.



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**.



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**.

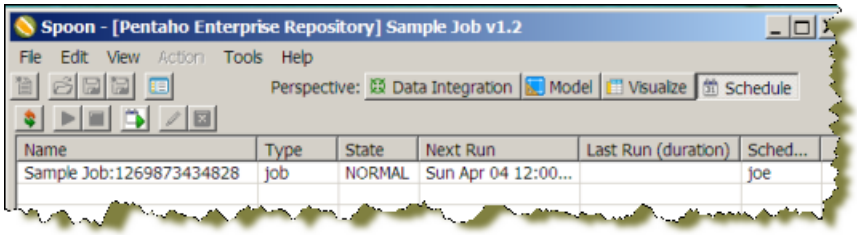
4. Under the **Repeat** section, select the **Weekly** option. Enable the **Sunday** check box.

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

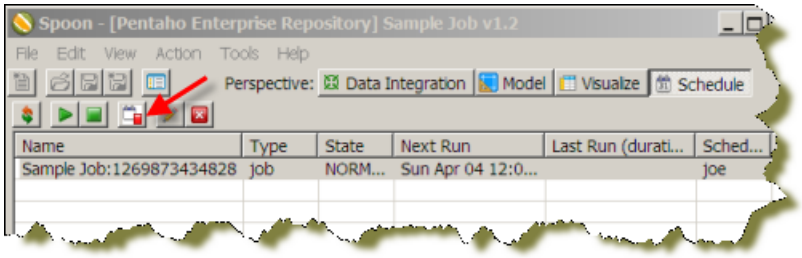
6. Click **OK** to complete your schedule.

Note: The scheduler includes full support for Pentaho Data Integration's parameters, arguments, and variables. For more detailed information on scheduling options, please refer to the *Pentaho Data Integration User Guide* found in the Pentaho InfoCenter.

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.



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.

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 the tab containing the sample transformation you just created.
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.

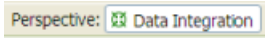
COUNTRY	SALES
Australia	630,623
Austria	202,063
Belgium	108,413
Canada	224,079
Denmark	245,637
Finland	329,582
France	1,110,917
Germany	220,472
Ireland	57,756
Italy	374,674
Japan	188,168
Norway	307,464
Philippines	94,016
Singapore	288,488
Spain	1,215,687
Sweden	210,014
Switzerland	117,714
UK	478,880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

Note: Immediately you can see that there is another problem with the quality of the data. 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.

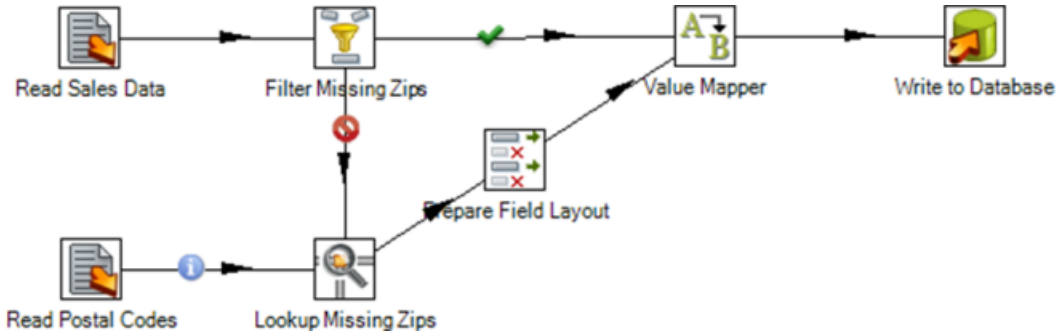
Correcting the Data Quality Issue

Follow the instructions below to correct the data quality issue:

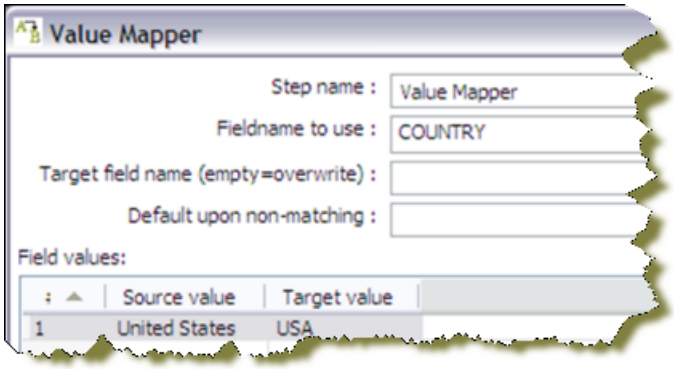
1. Click on the **Data Integration** perspective in the main toolbar.



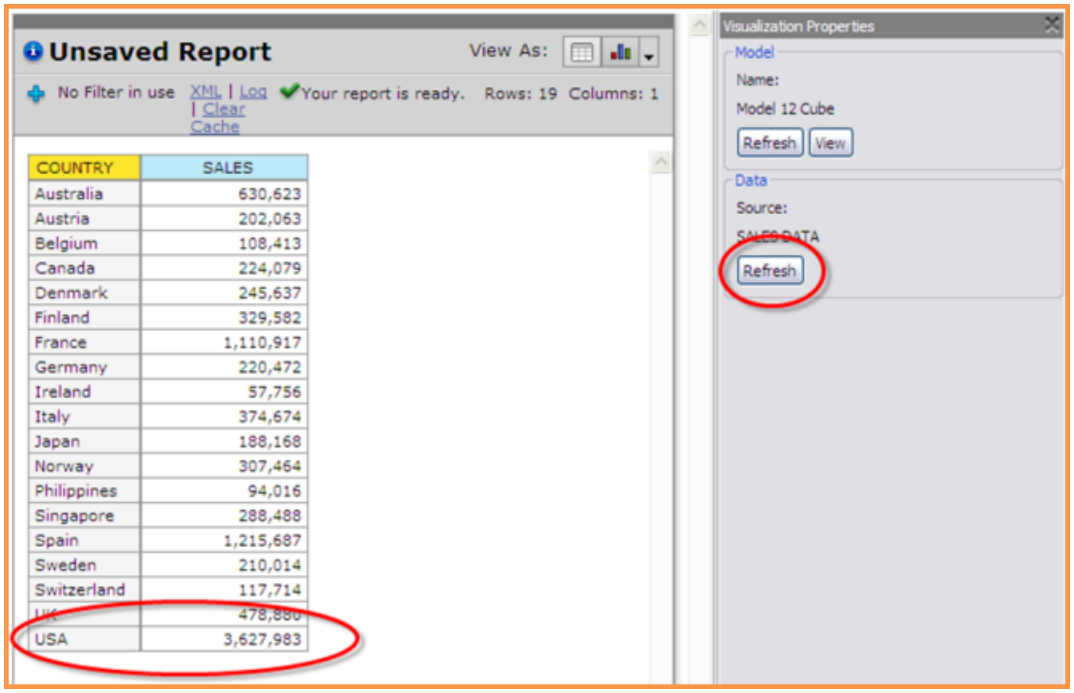
2. Right-click the **Table output** step from the flow and choose **Detach step**. Repeat this process to detach the second hop.
3. Expand the **Transform** folder in the Design Palette and add a **Value Mapper** step to the transformation.
4. Draw a hop from the **Filter Missing Zips** (Filter rows) 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.



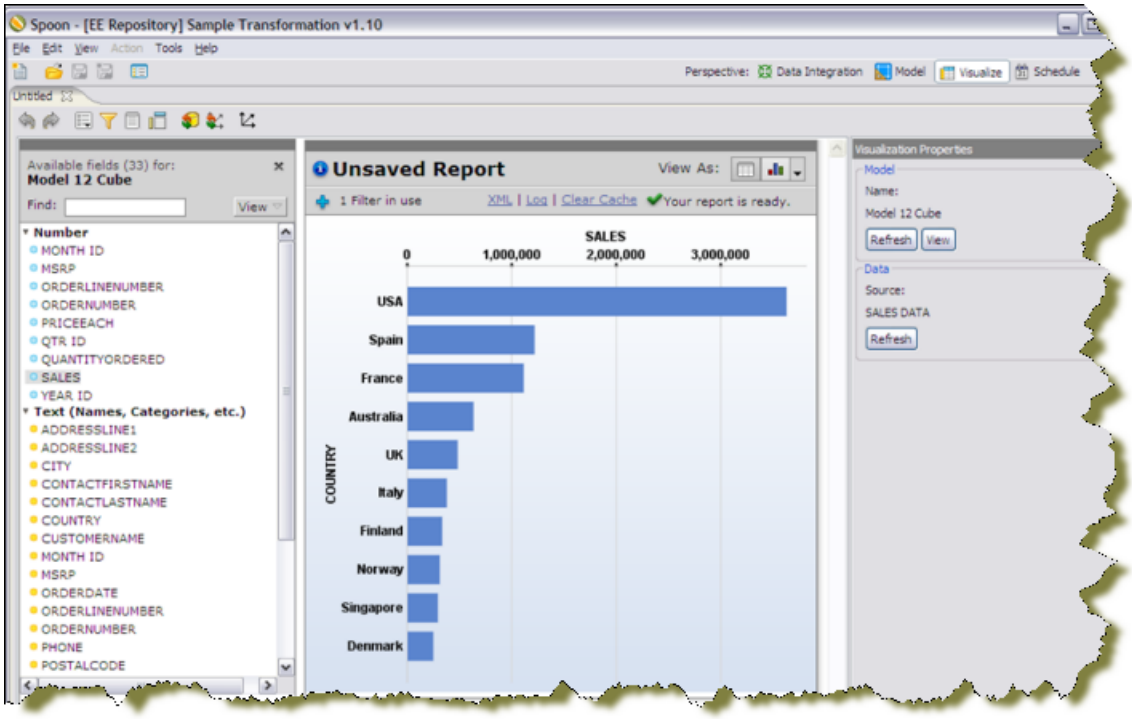
10. Save and run the transformation.
11. Click **Visualize** in the main toolbar.
12. Click the **Clear Cache** link at the top of the report.
13. Click **Refresh** under the data section of the **Visualization Properties** panel and notice that the data has been cleansed.



Creating a Top Ten Countries by Sales Chart

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

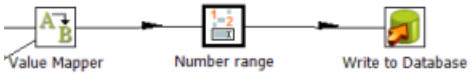
1. Right-click the **COUNTRY** header and select **Top 10**, and so on..
2. Confirm that the default settings are set to return the Top 10 **COUNTRY** members by the **SALES** measure. Click **OK**.
3. Click  (chart) and select **Stacked Bar** to change the visualization to a bar chart.



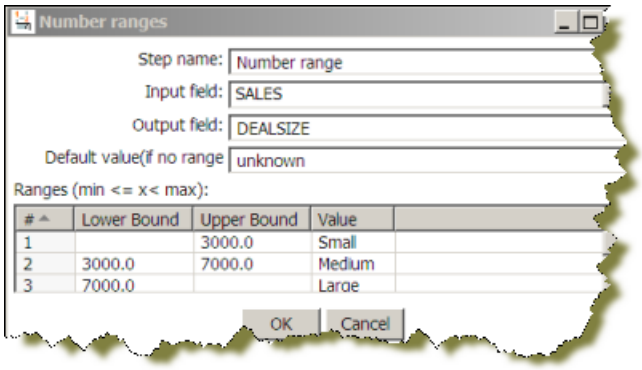
Breaking Down Your Chart by Deal Size

Your source data does not contain an attribute for Deal Size, 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 between the **Value Mapper** step and the **Write to Database** (Table Output) step. Click **Yes** to split the hop.



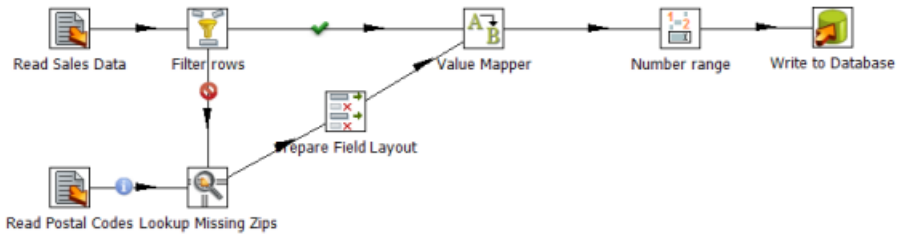
3. Double-click **Number range** to open its edit properties dialog box.
4. Choose the **SALES** field as your Input field.
5. Type **DEAL SIZE** as the name for the **Output** field.
6. In the **Ranges** table, define number ranges as shown in the example below. Click **OK**.



Note: Because this step will be adding 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.

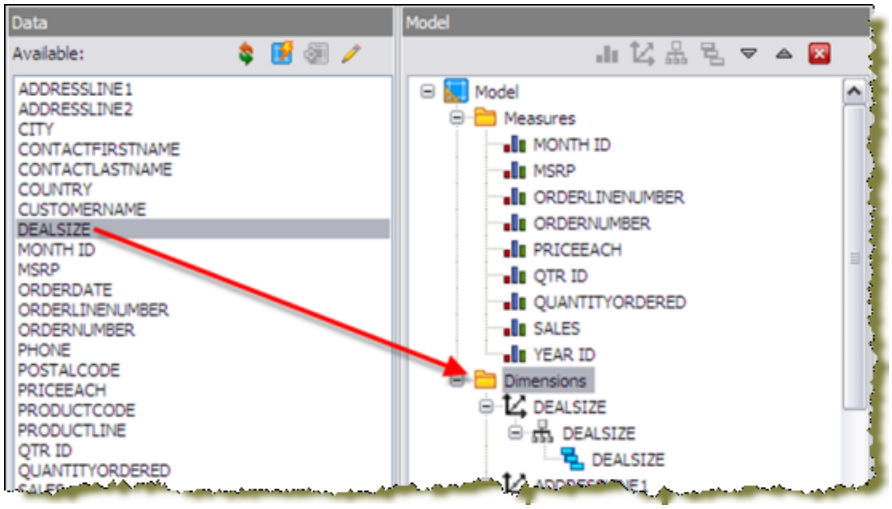
10. Save and run your transformation to re-populate your target database.



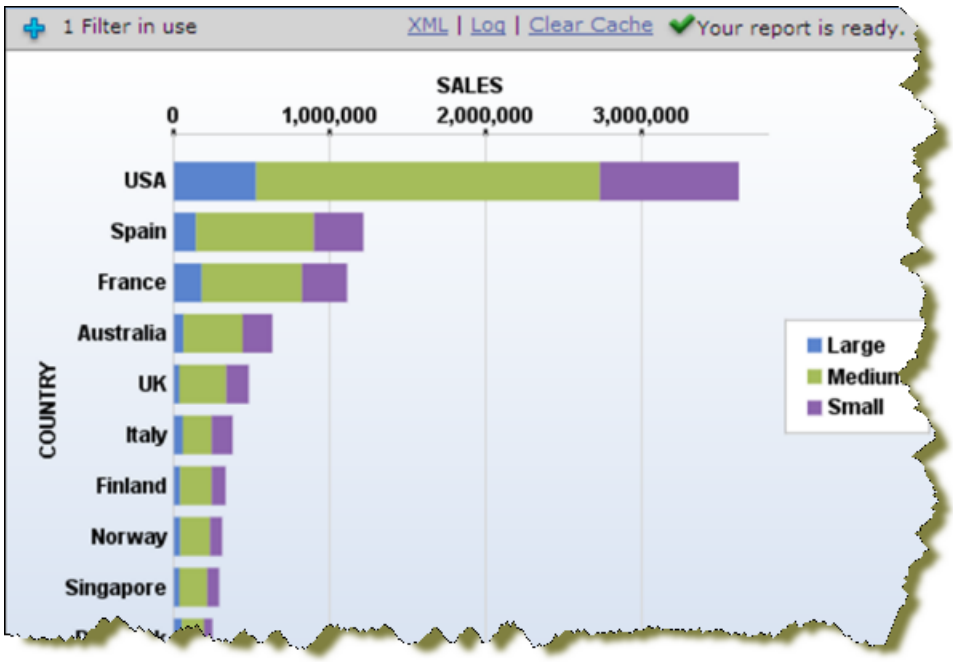
Wrapping it Up

Follow 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.



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


1 Filter in use
XML | Log | Clear Cache
Your report is ready.
SALES
0
1,000,000
2,000,000
3,000,000
COUNTRY
USA
Spain
France
Australia
UK
Italy
Finland
Norway
Singapore
Large
Medium
Small

Pentaho Data Integration and Big Data

Getting Started

Pentaho's big data support covers a broad spectrum of big data sources including Hadoop, NoSQL databases, and other high-performance analytic databases.

Pentaho Big Data components are open source in order to better function within the Hadoop open source ecosystem. Pentaho has put all of the Hadoop and NoSQL components into open source to make Kettle be the best and most pervasive ETL engine in the Big Data space.

To further Kettle adoption within the Hadoop community, Pentaho decided to move the Kettle open source license from GNU Lesser General Public License (LGPL) to the more permissive Apache license. This addresses issues of restrictions applied to a derivative work based on combining Kettle with Hadoop.

Using Hadoop

The Pentaho Data Integration (Kettle) client comes pre-configured for Apache Hadoop 0.20.2. If you are using this distribution and version, no further configuration is required after downloading and extracting Kettle.

Getting Started with Hadoop

To configure Kettle for a different version of Hadoop, you first should delete the core JAR file in `$PDI_HOME/libext/pentaho`, and replace it with the JAR file from your cluster.

For example, if you are using Cloudera CDHu2, you would copy `$HADOOP_HOME/hadoop-core-0.20.2-cdh3u2.jar` to `$PDI_HOME/libext/pentaho`

For certain Hadoop distributions or versions, for example Hadoop 0.20.205, you also need to have Apache Commons Configuration included in your set of PDI libraries. To do this, copy `commons-configuration-1.7.jar` to `$PDI_HOME/libext/commons`.

For other Hadoop distributions or versions, for example Cloudera CDH3 Update 3, you also need to copy the JAR file `$HADOOP_HOME/lib/guava-r09-jarjar.jar` to `$PRD_HOME/lib`.

Loading Data Into Hadoop's Distributed File System (HDFS)

In this guide you will learn how to copy local files into HDFS using PDI's graphical design tool, Spoon. You can use this tool to put files into the HDFS from many different sources.

In order to follow this guide you should already have Hadoop and Pentaho Data Integration, and both tools should be running.

Create a job to load files into HDFS using these steps:

1. **Create a new Job**: Select `File` -> `New` -> `Job` from the menu toolbar, or click the New File icon, then select `Job`.
2. **Add a Start step**: Find the Start step within the Design Palette underthe General section. Drag it to the Job Canvas.
3. **Add a Hadoop Copy Files step**: Find the Hadoop Copy Files step within the Design Palette, under the Big Data section. Drag the job entry onto the Job Canvas.
4. **Connect the Start and Copy Files Job Entries**: Hover the mouse over the Start step and a tooltip will appear. Click on the output connector (the green arrow pointing to the right), then drag the connector arrow to the Hadoop Copy Files step.
5. **Edit the Copy Files Job Entry**: Double-click the Hadoop Copy Files step to edit its properties. Next, enter the following information:
   a) `File/Folder source(s)`
   b) `File/Folder destination(s)`
   c) `Wildcard (RegExp)`
   d) Click the `Add` button to add the above entries to the list of files to copy.
   e) Check the `Create destination folder` option to ensure the weblogs folder is created in HDFS the first time the job is executed.
   f) Click `OK` to close the window.
6. **Save the Job**: Choose `File` -> `Save as...` from the menu, then save the transformation.

7. **Run the Job**: Choose `Action` -> `Run` from the menu (or click on the green run button on the Job Toolbar). An `Execute a job` window will open. Click on the `Launch` button. An `Execution Results` panel will open at the bottom of the PDI window and it will show you the progress of the job as it runs. After a few seconds the job should finish successfully. Results can be viewed in the `Job metrics` tab. If there were errors, the job step that failed will be highlighted in red, and you can read error messages in the `Logging` tab.
8. **Check Hadoop**: Run the following command:`hadoop fs -ls /user/pdi/[filename]`
-rwxrwxrwx 3 demo demo 77908174 2011-12-28 07:16 /user/pdi/[filename]

Using MapReduce

These instructions are specific to the MapR distribution of Hadoop.

Getting Started with MapReduce

Overview:

1. The MapR native libraries for your architecture must be added to the `java.library.path`.
2. MapR Hadoop Configuration directory needs to be on the classpath.
3. MapR Hadoop Core library must be on the classpath.

Configure Pentaho Data Integration Client for MapR

The following are PDI client configuration instructions for MapR:

1. `For all architectures`:
   a) Update the `$PDI_HOME/launcher/launcher.properties` with the attached launcher.properties
   b) Delete `$PDI_HOME/libext/pentaho/hadoop-0.20.2-core.jar`.
   c) Copy `$MAPR_HOME/hadoop/hadoop-0.20.2/lib/hadoop-0.20.2-dev-core.jar` into `$PDI_HOME/libext/bigdata`.
   d) Copy `$MAPR_HOME/hadoop/hadoop-0.20.2/lib/maprfs-0.1.jar` into `$PDI_HOME/libext/bigdata`.
2. `For Linux x64 systems`:
   a) Update the `$PDI_HOME/spoon.sh` with the attached spoon.sh
   b) Update the `$PDI_HOME/pan.sh` with the attached pan.sh
   c) Update the `$PDI_HOME/kitchen.sh` with the attached kitchen.sh
   d) Update the `$PDI_HOME/carte.sh` with the attached carte.sh
3. `For Mac OS X 64-bit systems`:
   a) Update the Data Integration `64-bit.app/Content/Info.plist` with the attached Info.plist

Configure Pentaho Report Designer for MapR

The following are PRD configuration instructions for MapR:

1. Delete $PRD_HOME/lib/jdbc/hadoop-0.20.2-core.jar
2. Copy $MAPR_HOME/hadoop/hadoop-0.20.2/lib/hadoop-0.20.2-dev-core.jar into $PRD_HOME/lib
3. Copy $MAPR_HOME/hadoop/hadoop-0.20.2/lib/maprfs-0.1.jar into $PRD_HOME/lib
4. `For Linux x64:`
   a) Add `-Djava.library.path=/opt/mapr/hadoop/hadoop-0.20.2/lib/native/Linux-amd64-64` to the last line in `$PRD_HOME/report-designer.sh`.
5. `For MacOS`:
   a) Add `-Djava.library.path=/opt/mapr/hadoop/hadoop-0.20.2/lib/native/Mac_OS_X-x86_64-64` to the VMOptions entry in `$PRD_HOME/Pentaho\Report\Designer.app/Contents/Info.plist`

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/*.

Troubleshooting

This section contains known problems and solutions relating to the procedures covered in this guide.

I don't know what the default login is for the DI Server, Enterprise Console, and/or Carte

For the DI Server administrator, it's username **admin** and password **secret**.

For Enterprise Console administrator, it's username **admin** and password **password**.

For Carte, it's username **cluster** and password **cluster**.

Be sure to change these to new values in your production environment.

Note: DI Server users are not the same as BI Server users.